



등록특허 10-2344232



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2021년12월27일
(11) 등록번호 10-2344232
(24) 등록일자 2021년12월23일

- (51) 국제특허분류(Int. Cl.)
H04N 19/93 (2014.01) *H04N 19/119* (2014.01)
H04N 19/176 (2014.01) *H04N 19/44* (2014.01)
H04N 19/50 (2014.01) *H04N 19/593* (2014.01)
- (52) CPC특허분류
H04N 19/93 (2015.01)
H04N 19/119 (2015.01)
- (21) 출원번호 10-2016-7027491
- (22) 출원일자(국제) 2015년03월25일
심사청구일자 2020년03월09일
- (85) 번역문제출일자 2016년10월04일
- (65) 공개번호 10-2016-0136338
- (43) 공개일자 2016년11월29일
- (86) 국제출원번호 PCT/US2015/022468
- (87) 국제공개번호 WO 2015/148652
국제공개일자 2015년10월01일
- (30) 우선권주장
61/970,257 2014년03월25일 미국(US)
(뒷면에 계속)
- (56) 선행기술조사문헌
J.Chen et al., "Description of screen content coding technology proposal by Qualcomm," (JCTVC-Q0031), JCT-VC of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, (2014.03.27.)
(뒷면에 계속)

전체 청구항 수 : 총 52 항

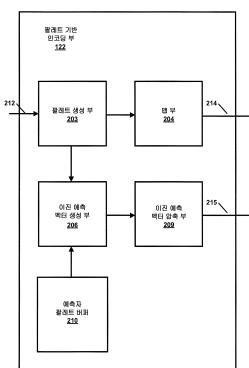
심사관 : 조우연

(54) 발명의 명칭 비디오 코딩을 위한 런 길이 코드를 이용한 팔레트 예측자 시그널링

(57) 요 약

팔레트 기반 비디오 코딩을 위한 팔레트를 예측하기 위한 이진 예측 벡터를 인코딩하는 기법들이 설명된다. 하나의 예에서, 비디오를 디코딩하는 방법이 비디오 데이터의 현재 블록에 대한 인코딩된 이진 예측 벡터를 수신하는 단계, 런-길이 디코딩 기법을 사용하여 인코딩된 이진 예측 벡터를 디코딩하는 단계, 이진 예측 벡터에 기

(뒷면에 계속)

대 표 도 - 도4

초하여 비디오 데이터의 현재 블록에 대한 팔레트를 생성하는 단계로서, 이전 예측 벡터는 이전에 사용된 팔레트 엔트리들이 비디오 데이터의 현재 블록에 대한 팔레트에 재사용되는지의 여부를 나타내는 엔트리들을 포함하는, 상기 비디오 데이터의 현재 블록에 대한 팔레트를 생성하는 단계, 및 그 팔레트를 사용하여 비디오 데이터의 현재 블록을 디코딩하는 단계를 포함한다.

(52) CPC특허분류

HO4N 19/176 (2015.01)*HO4N 19/44* (2015.01)*HO4N 19/50* (2015.01)*HO4N 19/593* (2015.01)

(72) 발명자

세례진 바딤

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

조쉬 라잔 랙스맨

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

솔레 로할스 호텔

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

제우 평

미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775

(56) 선행기술조사문헌

US20120057799 A1

US20160337649A1

KR1020090005080 A

KR1020110037898 A

(30) 우선권주장

61/981,105 2014년04월17일 미국(US)

62/002,668 2014년05월23일 미국(US)

62/015,327 2014년06월20일 미국(US)

62/018,461 2014년06월27일 미국(US)

62/041,119 2014년08월24일 미국(US)

14/667,411 2015년03월24일 미국(US)

명세서

청구범위

청구항 1

비디오 데이터를 디코딩하는 방법으로서,

비디오 데이터의 현재 블록에 대한 인코딩된 이진 예측 벡터를 수신하는 단계;

런-길이 디코딩 기법을 사용하여 상기 인코딩된 이진 예측 벡터를 디코딩하는 단계;

상기 이진 예측 벡터에 기초하여 상기 비디오 데이터의 현재 블록에 대한 팔레트를 생성하는 단계로서, 상기 이진 예측 벡터는 이전에 사용된 팔레트 엔트리들이 상기 비디오 데이터의 현재 블록에 대한 상기 팔레트에 재사용되는지의 여부를 나타내는 엔트리들을 포함하는, 상기 비디오 데이터의 현재 블록에 대한 팔레트를 생성하는 단계; 및

상기 팔레트를 사용하여 상기 비디오 데이터의 현재 블록을 디코딩하는 단계를 포함하는, 비디오 데이터 디코딩 방법.

청구항 2

제 1 항에 있어서,

상기 인코딩된 이진 예측 벡터를 디코딩하는 단계는 상기 런-길이 디코딩 기법에 의해 생성된 런-길이 시퀀스에 대해 지수-골름 디코딩 기법을 사용하여 상기 인코딩된 이진 예측 벡터를 디코딩하는 단계를 포함하는, 비디오 데이터 디코딩 방법.

청구항 3

제 2 항에 있어서,

상기 지수-골름 디코딩 기법은 0차 지수-골름 디코딩 기법인, 비디오 데이터 디코딩 방법.

청구항 4

제 1 항에 있어서,

상기 인코딩된 이진 예측 벡터를 디코딩하는 단계는,

상기 런-길이 디코딩 기법과 상기 이진 예측 벡터의 말단-포지션을 나타내는 예약된 런-길이 값 L 을 사용하여 상기 인코딩된 이진 예측 벡터를 디코딩하는 단계를 포함하는, 비디오 데이터 디코딩 방법.

청구항 5

제 4 항에 있어서,

상기 예약된 런-길이 값 L 은 1인, 비디오 데이터 디코딩 방법.

청구항 6

제 1 항에 있어서,

상기 인코딩된 이진 예측 벡터를 디코딩하는 단계는,

상기 런-길이 디코딩 기법, 상기 이진 예측 벡터의 말단-포지션을 나타내는 예약된 런-길이 값 (L), 및 최대 팔레트 사이즈를 사용하여 상기 인코딩된 이진 예측 벡터를 디코딩하는 단계로서, 이전에 사용된 팔레트 엔트리들이 상기 비디오 데이터의 현재 블록에 대한 상기 팔레트에 재사용됨을 나타내는 상기 이진 예측 벡터에서의 엔트리들의 총 수가 상기 최대 팔레트 사이즈와 동일하다면 상기 예약된 런-길이 값 L 은 사용되지 않는, 상기 인코딩된 이진 예측 벡터를 디코딩하는 단계를 포함하는, 비디오 데이터 디코딩 방법.

청구항 7

제 1 항에 있어서,

상기 인코딩된 이진 예측 벡터를 디코딩하는 단계는,

상기 런-길이 디코딩 기법과 상기 이진 예측 벡터의 말단-포지션을 나타내는 예약된 런-길이 값 L 을 사용하여 상기 인코딩된 이진 예측 벡터를 디코딩하는 단계로서, 상기 인코딩된 이진 예측 벡터에서의 마지막 런-길이가 영의 런-길이를 나타내지 않는다면 상기 예약된 런-길이 값 L 은 사용되지 않는, 상기 인코딩된 이진 예측 벡터를 디코딩하는 단계를 포함하는, 비디오 데이터 디코딩 방법.

청구항 8

제 1 항에 있어서,

상기 이진 예측 벡터에 기초하여 상기 비디오 데이터의 현재 블록에 대한 팔레트를 생성하는 단계는,

버퍼로부터, 상기 이진 예측 벡터에 의해 상기 팔레트에 재사용되고 있는 것으로서 나타내어진 이전에 사용된 팔레트 엔트리들을 상기 팔레트 속으로 복사하는 단계; 및

상기 팔레트에 복사된 이전에 사용된 팔레트 엔트리들의 수가 최대 팔레트 사이즈 미만인 경우, 추가적인 팔레트 엔트리들을 수신하는 단계를 포함하는, 비디오 데이터 디코딩 방법.

청구항 9

제 8 항에 있어서,

상기 버퍼에 비디오 데이터의 하나 이상의 이전에 디코딩된 블록들에 대한 상기 이전에 사용된 팔레트 엔트리들을 저장하는 단계를 더 포함하는, 비디오 데이터 디코딩 방법.

청구항 10

제 9 항에 있어서,

상기 버퍼에 저장된 상기 이전에 사용된 팔레트 엔트리들의 중복 엔트리들을 제거하는 단계를 더 포함하는, 비디오 데이터 디코딩 방법.

청구항 11

제 9 항에 있어서,

비디오 데이터의 상기 하나 이상의 이전에 디코딩된 블록들에 대한 상기 이전에 사용된 팔레트 엔트리들은, 상기 비디오 데이터의 현재 블록 상측의 화소들의 라인에 대한 이전에 사용된 팔레트 엔트리들과 상기 비디오 데이터의 현재 블록 좌측의 화소들의 라인에 대한 이전에 사용된 팔레트 엔트리들을 포함하는, 비디오 데이터 디코딩 방법.

청구항 12

제 1 항에 있어서,

상기 런-길이 디코딩 기법은 영들의 런-길이를 코딩하는 단계를 포함하는, 비디오 데이터 디코딩 방법.

청구항 13

비디오 데이터를 인코딩하는 방법으로서,

상기 비디오 데이터의 현재 블록에 대한 팔레트를 생성하는 단계;

상기 비디오 데이터의 현재 블록에 대한 상기 팔레트에 대한 이진 예측 벡터를 생성하는 단계로서, 상기 이진 예측 벡터는 이전에 사용된 팔레트 엔트리들이 상기 비디오 데이터의 현재 블록에 대한 상기 팔레트에 재사용되는지의 여부를 나타내는 엔트리들을 포함하는, 상기 이진 예측 벡터를 생성하는 단계;

런-길이 인코딩 기법을 사용하여 상기 이진 예측 벡터를 인코딩하는 단계; 및

상기 팔레트를 사용하여 상기 비디오 데이터의 현재 블록을 인코딩하는 단계를 포함하는, 비디오 데이터 인코딩 방법.

청구항 14

제 13 항에 있어서,

상기 이진 예측 벡터를 인코딩하는 단계는 런-길이 디코딩 기법에 의해 생성된 런-길이 시퀀스에 대해 지수-골룸 인코딩 기법을 사용하여 상기 이진 예측 벡터를 인코딩하는 단계를 포함하는, 비디오 데이터 인코딩 방법.

청구항 15

제 14 항에 있어서,

상기 지수-골룸 인코딩 기법은 0차 지수-골룸 인코딩 기법인, 비디오 데이터 인코딩 방법.

청구항 16

제 13 항에 있어서,

상기 이진 예측 벡터를 인코딩하는 단계는,

상기 런-길이 인코딩 기법과 상기 이진 예측 벡터의 말단-포지션을 나타내는 예약된 런-길이 값 L 을 사용하여 상기 이진 예측 벡터를 인코딩하는 단계를 포함하는, 비디오 데이터 인코딩 방법.

청구항 17

제 16 항에 있어서,

상기 예약된 런-길이 값 L 은 1인, 비디오 데이터 인코딩 방법.

청구항 18

제 13 항에 있어서,

상기 이진 예측 벡터를 인코딩하는 단계는,

상기 런-길이 인코딩 기법, 상기 이진 예측 벡터의 말단-포지션을 나타내는 예약된 런-길이 값 (L), 및 최대 팔레트 사이즈를 사용하여 상기 이진 예측 벡터를 인코딩하는 단계로서, 이전에 사용된 팔레트 엔트리들이 상기 비디오 데이터의 현재 블록에 대한 상기 팔레트에 재사용됨을 나타내는 상기 이진 예측 벡터에서의 엔트리들의 총 수가 상기 최대 팔레트 사이즈와 동일하다면 상기 예약된 런-길이 값 L 은 사용되지 않는, 상기 이진 예측 벡터를 인코딩하는 단계를 포함하는, 비디오 데이터 인코딩 방법.

청구항 19

제 13 항에 있어서,

상기 이진 예측 벡터를 인코딩하는 단계는,

상기 런-길이 인코딩 기법과 상기 이진 예측 벡터의 말단-포지션을 나타내는 예약된 런-길이 값 L 을 사용하여 상기 이진 예측 벡터를 인코딩하는 단계로서, 상기 인코딩된 이진 예측 벡터에서의 마지막 런-길이가 영의 런-길이를 나타내지 않는다면 상기 예약된 런-길이 값 L 은 사용되지 않는, 상기 이진 예측 벡터를 인코딩하는 단계를 포함하는, 비디오 데이터 인코딩 방법.

청구항 20

제 13 항에 있어서,

버퍼에 비디오 데이터의 하나 이상의 이전에 인코딩된 블록들에 대한 상기 이전에 사용된 팔레트 엔트리들을 저장하는 단계를 더 포함하는, 비디오 데이터 인코딩 방법.

청구항 21

제 20 항에 있어서,

상기 베퍼에 저장된 상기 이전에 사용된 팔레트 엔트리들의 중복 엔트리를 제거하는 단계를 더 포함하는, 비디오 데이터 인코딩 방법.

청구항 22

제 20 항에 있어서,

비디오 데이터의 상기 하나 이상의 이전에 인코딩된 블록들에 대한 상기 이전에 사용된 팔레트 엔트리들은, 상기 비디오 데이터의 현재 블록 상측의 화소들의 라인에 대한 이전에 사용된 팔레트 엔트리들과 상기 비디오 데이터의 현재 블록 좌측의 화소들의 라인에 대한 이전에 사용된 팔레트 엔트리들을 포함하는, 비디오 데이터 인코딩 방법.

청구항 23

제 13 항에 있어서,

상기 런-길이 인코딩 기법은 영들의 런-길이를 코딩하는 단계를 포함하는, 비디오 데이터 인코딩 방법.

청구항 24

비디오 데이터를 디코딩하도록 구성되는 장치로서,

상기 비디오 데이터를 저장하도록 구성되는 메모리; 및

상기 메모리와 통신하고 있는 비디오 디코더를 포함하며,

상기 비디오 디코더는,

상기 비디오 데이터의 현재 블록에 대한 인코딩된 이진 예측 벡터를 수신하고;

런-길이 디코딩 기법을 사용하여 상기 인코딩된 이진 예측 벡터를 디코딩하고;

상기 이진 예측 벡터에 기초하여 상기 비디오 데이터의 현재 블록에 대한 팔레트를 생성하는 것으로서, 상기 이진 예측 벡터는 이전에 사용된 팔레트 엔트리들이 상기 비디오 데이터의 현재 블록에 대한 상기 팔레트에 재사용되는지의 여부를 나타내는 엔트리들을 포함하는, 상기 비디오 데이터의 현재 블록에 대한 팔레트를 생성하고; 및

상기 팔레트를 사용하여 상기 비디오 데이터의 현재 블록을 디코딩하도록

구성되는, 비디오 데이터를 디코딩하도록 구성되는 장치.

청구항 25

제 24 항에 있어서,

상기 인코딩된 이진 예측 벡터를 디코딩하기 위해, 상기 비디오 디코더는 또한, 상기 런-길이 디코딩 기법에 의해 생성된 런-길이 시퀀스에 대해 지수-골롬 디코딩 기법을 사용하여 상기 인코딩된 이진 예측 벡터를 디코딩하도록 구성되는, 비디오 데이터를 디코딩하도록 구성되는 장치.

청구항 26

제 25 항에 있어서,

상기 지수-골롬 디코딩 기법은 0차 지수-골롬 디코딩 기법인, 비디오 데이터를 디코딩하도록 구성되는 장치.

청구항 27

제 24 항에 있어서,

상기 인코딩된 이진 예측 벡터를 디코딩하기 위해, 상기 비디오 디코더는 또한,

상기 런-길이 디코딩 기법과 상기 이진 예측 벡터의 말단-포지션을 나타내는 예약된 런-길이 값 L 을 사용하여

상기 인코딩된 이진 예측 벡터를 디코딩하도록 구성되는, 비디오 데이터를 디코딩하도록 구성되는 장치.

청구항 28

제 27 항에 있어서,

상기 예약된 런-길이 값 L 은 1인, 비디오 데이터를 디코딩하도록 구성되는 장치.

청구항 29

제 24 항에 있어서,

상기 인코딩된 이진 예측 벡터를 디코딩하기 위해, 상기 비디오 디코더는 또한,

상기 런-길이 디코딩 기법, 상기 이진 예측 벡터의 말단-포지션을 나타내는 예약된 런-길이 값 (L), 및 최대 팔레트 사이즈를 사용하여 상기 인코딩된 이진 예측 벡터를 디코딩하도록 구성되며,

이전에 사용된 팔레트 엔트리들이 상기 비디오 데이터의 현재 블록에 대한 상기 팔레트에 재사용됨을 나타내는 상기 이진 예측 벡터에서의 엔트리들의 총 수가 상기 최대 팔레트 사이즈와 동일하다면 상기 예약된 런-길이 값 L 은 사용되지 않는, 비디오 데이터를 디코딩하도록 구성되는 장치.

청구항 30

제 24 항에 있어서,

상기 인코딩된 이진 예측 벡터를 디코딩하기 위해, 상기 비디오 디코더는 또한,

상기 런-길이 디코딩 기법과 상기 이진 예측 벡터의 말단-포지션을 나타내는 예약된 런-길이 값 L 을 사용하여 상기 인코딩된 이진 예측 벡터를 디코딩하도록 구성되며,

상기 인코딩된 이진 예측 벡터에서의 마지막 런-길이가 영의 런-길이를 나타내지 않는다면 상기 예약된 런-길이 값 L 은 사용되지 않는, 비디오 데이터를 디코딩하도록 구성되는 장치.

청구항 31

제 24 항에 있어서,

상기 이진 예측 벡터에 기초하여 상기 비디오 데이터의 현재 블록에 대한 상기 팔레트를 생성하기 위해, 상기 비디오 디코더는 또한,

버퍼로부터, 상기 이진 예측 벡터에 의해 상기 팔레트에 재사용되고 있는 것으로서 나타내어진 이전에 사용된 팔레트 엔트리들을 상기 팔레트 속으로 복사하고; 및

상기 팔레트에 복사된 이전에 사용된 팔레트 엔트리들의 수가 최대 팔레트 사이즈 미만인 경우, 추가적인 팔레트 엔트리들을 수신하도록 구성되는, 비디오 데이터를 디코딩하도록 구성되는 장치.

청구항 32

제 31 항에 있어서,

상기 비디오 디코더는 또한,

상기 버퍼에 비디오 데이터의 하나 이상의 이전에 디코딩된 블록들에 대한 상기 이전에 사용된 팔레트 엔트리들을 저장하도록 구성되는, 비디오 데이터를 디코딩하도록 구성되는 장치.

청구항 33

제 32 항에 있어서,

상기 비디오 디코더는 또한,

상기 버퍼에 저장된 상기 이전에 사용된 팔레트 엔트리들의 중복 엔트리들을 제거하도록 구성되는, 비디오 데이터를 디코딩하도록 구성되는 장치.

청구항 34

제 32 항에 있어서,

비디오 데이터의 상기 하나 이상의 이전에 디코딩된 블록들에 대한 상기 이전에 사용된 팔레트 엔트리들은, 상기 비디오 데이터의 현재 블록 상측의 화소들의 라인에 대한 이전에 사용된 팔레트 엔트리들과 상기 비디오 데이터의 현재 블록 좌측의 화소들의 라인에 대한 이전에 사용된 팔레트 엔트리들을 포함하는, 비디오 데이터를 디코딩하도록 구성되는 장치.

청구항 35

제 24 항에 있어서,

상기 런-길이 디코딩 기법은 영들의 런-길이를 코딩하는 것을 포함하는, 비디오 데이터를 디코딩하도록 구성되는 장치.

청구항 36

제 24 항에 있어서,

상기 비디오 데이터의 상기 디코딩된 현재 블록을 포함하는 픽처를 디스플레이하도록 구성되는 디스플레이를 더 포함하는, 비디오 데이터를 디코딩하도록 구성되는 장치.

청구항 37

비디오 데이터를 인코딩하도록 구성되는 장치로서,

상기 비디오 데이터를 저장하도록 구성되는 메모리; 및

상기 메모리와 통신하고 있는 비디오 인코더를 포함하며,

상기 비디오 인코더는,

상기 비디오 데이터의 현재 블록에 대한 팔레트를 생성하고;

상기 비디오 데이터의 현재 블록에 대한 상기 팔레트에 대한 이진 예측 벡터를 생성하는 것으로서, 상기 이진 예측 벡터는 이전에 사용된 팔레트 엔트리들이 상기 비디오 데이터의 현재 블록에 대한 상기 팔레트에 재사용되는지의 여부를 나타내는 엔트리들을 포함하는, 상기 비디오 데이터의 현재 블록에 대한 상기 팔레트에 대한 이진 예측 벡터를 생성하고;

런-길이 인코딩 기법을 사용하여 상기 이진 예측 벡터를 인코딩하고; 및

상기 팔레트를 사용하여 상기 비디오 데이터의 현재 블록을 인코딩하도록 구성되는, 비디오 데이터를 인코딩하도록 구성되는 장치.

청구항 38

제 37 항에 있어서,

상기 이진 예측 벡터를 인코딩하기 위해, 상기 비디오 인코더는 또한,

런-길이 디코딩 기법에 의해 생성된 런-길이 시퀀스에 대해 지수-콜롬 인코딩 기법을 사용하여 상기 이진 예측 벡터를 인코딩하도록 구성되는, 비디오 데이터를 인코딩하도록 구성되는 장치.

청구항 39

제 38 항에 있어서,

상기 지수-콜롬 인코딩 기법은 0차 지수-콜롬 인코딩 기법인, 비디오 데이터를 인코딩하도록 구성되는 장치.

청구항 40

제 37 항에 있어서,

상기 이진 예측 벡터를 인코딩하기 위해, 상기 비디오 인코더는 또한,

상기 런-길이 인코딩 기법과 상기 이진 예측 벡터의 말단-포지션을 나타내는 예약된 런-길이 값 L 을 사용하여 상기 이진 예측 벡터를 인코딩하도록 구성되는, 비디오 데이터를 인코딩하도록 구성되는 장치.

청구항 41

제 40 항에 있어서,

상기 예약된 런-길이 값 L 은 1인, 비디오 데이터를 인코딩하도록 구성되는 장치.

청구항 42

제 37 항에 있어서,

상기 이진 예측 벡터를 인코딩하기 위해, 상기 비디오 인코더는 또한,

상기 런-길이 인코딩 기법, 상기 이진 예측 벡터의 말단-포지션을 나타내는 예약된 런-길이 값 (L), 및 최대 팔레트 사이즈를 사용하여 상기 이진 예측 벡터를 인코딩하도록 구성되며,

이전에 사용된 팔레트 엔트리들이 상기 비디오 데이터의 현재 블록에 대한 상기 팔레트에 재사용됨을 나타내는 상기 이진 예측 벡터에서의 엔트리들의 총 수가 상기 최대 팔레트 사이즈와 동일하다면 상기 예약된 런-길이 값 L 은 사용되지 않는, 비디오 데이터를 인코딩하도록 구성되는 장치.

청구항 43

제 37 항에 있어서,

상기 이진 예측 벡터를 인코딩하기 위해, 상기 비디오 인코더는 또한,

상기 런-길이 인코딩 기법과 상기 이진 예측 벡터의 말단-포지션을 나타내는 예약된 런-길이 값 L 을 사용하여 상기 이진 예측 벡터를 인코딩하도록 구성되며,

상기 인코딩된 이진 예측 벡터에서의 마지막 런-길이가 영의 런-길이를 나타내지 않는다면 상기 예약된 런-길이 값 L 은 사용되지 않는, 비디오 데이터를 인코딩하도록 구성되는 장치.

청구항 44

제 37 항에 있어서,

상기 비디오 인코더는 또한,

버퍼에 비디오 데이터의 하나 이상의 이전에 인코딩된 블록들에 대한 상기 이전에 사용된 팔레트 엔트리들을 저장하도록 구성되는, 비디오 데이터를 인코딩하도록 구성되는 장치.

청구항 45

제 44 항에 있어서,

상기 비디오 인코더는 또한,

상기 버퍼에 저장된 상기 이전에 사용된 팔레트 엔트리들의 중복 엔트리들을 제거하도록 구성되는, 비디오 데이터를 인코딩하도록 구성되는 장치.

청구항 46

제 44 항에 있어서,

비디오 데이터의 상기 하나 이상의 이전에 인코딩된 블록들에 대한 상기 이전에 사용된 팔레트 엔트리들은, 상기 비디오 데이터의 현재 블록 상측의 화소들의 라인에 대한 이전에 사용된 팔레트 엔트리들과 상기 비디오 데이터의 현재 블록 좌측의 화소들의 라인에 대한 이전에 사용된 팔레트 엔트리들을 포함하는, 비디오 데이터를 인코딩하도록 구성되는 장치.

청구항 47

제 37 항에 있어서,

상기 런-길이 인코딩 기법은 영들의 런-길이를 코딩하는 것을 포함하는, 비디오 데이터를 인코딩하도록 구성되는 장치.

청구항 48

제 37 항에 있어서,

상기 비디오 데이터를 캡처하도록 구성되는 카메라를 더 포함하는, 비디오 데이터를 인코딩하도록 구성되는 장치.

청구항 49

비디오 데이터를 디코딩하도록 구성되는 장치로서,

비디오 데이터의 현재 블록에 대한 인코딩된 이진 예측 벡터를 수신하는 수단;

런-길이 디코딩 기법을 사용하여 상기 인코딩된 이진 예측 벡터를 디코딩하는 수단;

상기 이진 예측 벡터에 기초하여 상기 비디오 데이터의 현재 블록에 대한 팔레트를 생성하는 수단으로서, 상기 이진 예측 벡터는 이전에 사용된 팔레트 엔트리들이 상기 비디오 데이터의 현재 블록에 대한 상기 팔레트에 재사용되는지의 여부를 나타내는 엔트리들을 포함하는, 상기 비디오 데이터의 현재 블록에 대한 팔레트를 생성하는 수단; 및

상기 팔레트를 사용하여 상기 비디오 데이터의 현재 블록을 디코딩하는 수단을 포함하는, 비디오 데이터를 디코딩하도록 구성되는 장치.

청구항 50

비디오 데이터를 인코딩하도록 구성되는 장치로서,

상기 비디오 데이터의 현재 블록에 대한 팔레트를 생성하는 수단;

상기 비디오 데이터의 현재 블록에 대한 상기 팔레트에 대한 이진 예측 벡터를 생성하는 수단으로서, 상기 이진 예측 벡터는 이전에 사용된 팔레트 엔트리들이 상기 비디오 데이터의 현재 블록에 대한 상기 팔레트에 재사용되는지의 여부를 나타내는 엔트리들을 포함하는, 상기 이진 예측 벡터를 생성하는 수단;

런-길이 인코딩 기법을 사용하여 상기 이진 예측 벡터를 인코딩하는 수단; 및

상기 팔레트를 사용하여 상기 비디오 데이터의 현재 블록을 인코딩하는 수단을 포함하는, 비디오 데이터를 인코딩하도록 구성되는 장치.

청구항 51

명령들을 저장하는 컴퓨터 판독가능 저장 매체로서,

상기 명령들은, 실행되는 경우, 비디오 데이터를 디코딩하도록 구성되는 디바이스의 하나 이상의 프로세서들로 하여금,

상기 비디오 데이터의 현재 블록에 대한 인코딩된 이진 예측 벡터를 수신하게 하고;

런-길이 디코딩 기법을 사용하여 상기 인코딩된 이진 예측 벡터를 디코딩하게 하고;

상기 이진 예측 벡터에 기초하여 상기 비디오 데이터의 현재 블록에 대한 팔레트를 생성하게 하는 것으로서, 상기 이진 예측 벡터는 이전에 사용된 팔레트 엔트리들이 상기 비디오 데이터의 현재 블록에 대한 팔레트에 재사용되는지의 여부를 나타내는 엔트리들을 포함하는, 상기 비디오 데이터의 현재 블록에 대한 상기 팔레트를 생성하게 하고; 및

상기 팔레트를 사용하여 상기 비디오 데이터의 현재 블록을 디코딩하게 하는, 컴퓨터 판독가능 저장 매체.

청구항 52

명령들을 저장하는 컴퓨터 판독가능 저장 매체로서,

상기 명령들은, 실행되는 경우, 비디오 데이터를 인코딩하도록 구성되는 디바이스의 하나 이상의 프로세서들로 하여금,

상기 비디오 데이터의 현재 블록에 대한 팔레트를 생성하게 하고;

상기 비디오 데이터의 현재 블록에 대한 상기 팔레트에 대한 이진 예측 벡터를 생성하게 하는 것으로서, 상기 이진 예측 벡터는 이전에 사용된 팔레트 엔트리들이 상기 비디오 데이터의 현재 블록에 대한 상기 팔레트에 재사용되는지의 여부를 나타내는 엔트리들을 포함하는, 상기 비디오 데이터의 현재 블록에 대한 상기 팔레트에 대한 이진 예측 벡터를 생성하게 하고;

런-길이 인코딩 기법을 사용하여 상기 이진 예측 벡터를 인코딩하게 하고; 및

상기 팔레트를 사용하여 상기 비디오 데이터의 현재 블록을 인코딩하게 하는, 컴퓨터 판독가능 저장 매체.

발명의 설명

기술 분야

[0001]

본 출원은 2014년 3월 25일자로 출원된 미국 가 출원 제61/970,257호, 2014년 4월 17일 자로 출원된 미국 가 출원 제61/981,105호, 2014년 5월 23일자로 출원된 미국 가 출원 제62/002,668호, 2014년 6월 20일자로 출원된 미국 가 출원 제62/015,327호, 2014년 6월 27일자로 출원된 미국 가 출원 제62/018,461호, 및 2014년 8월 24일자로 출원된 미국 가 출원 제62/041,119호를 우선권 주장하며, 그것들의 각각의 전체 내용들은 참조로 본원에 통합된다.

[0002]

본 개시물은 비디오 인코딩 및 디코딩에 관한 것이다.

배경 기술

[0003]

디지털 비디오 능력들은 디지털 텔레비전들, 디지털 직접 브로드캐스트 시스템들, 무선 브로드캐스트 시스템들, 개인 정보 단말기들 (personal digital assistants, PDA들), 랩톱 또는 데스크톱 컴퓨터들, 태블릿 컴퓨터들, e-북 리더들, 디지털 카메라들, 디지털 레코딩 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 디바이스들, 비디오 게임 콘솔들, 셀룰러 또는 위성 무선 전화기들, 이른바 "스마트 폰들", 비디오 원격회의 디바이스들, 비디오 스트리밍 디바이스들 등을 포함한 넓은 범위의 디바이스들에 통합될 수 있다. 디지털 비디오 디바이스들은 MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, 파트 10, 고급 비디오 코딩 (Advanced Video Coding, AVC) 에 의해 규정된 표준들, 현재 개발중인 고 효율 비디오 코딩 (High Efficiency Video Coding, HEVC) 표준, 및 이러한 표준들의 확장본들에 기재된 것들과 같은 비디오 압축 기법들을 구현한다. 비디오 디바이스들은 이러한 비디오 압축 기법들을 구현하는 것에 의해 디지털 비디오 정보를 더 효율적으로 송신, 수신, 인코딩, 디코딩, 및/또는 저장할 수도 있다.

[0004]

비디오 압축 기법들은 공간적 (픽처 내) 예측 및/또는 시간적 (픽처 간) 예측을 수행하여 비디오 시퀀스들에 내재하는 리던던시를 감소시키거나 또는 제거한다. 블록 기반 비디오 코딩의 경우, 비디오 슬라이스 (즉, 비디오 프레임 또는 비디오 프레임의 부분) 가 비디오 블록들로 구획화될 수도 있다. 픽처의 인트라 코딩식 (intra-coded; I) 슬라이스에서의 비디오 블록들은 동일한 픽처의 이웃 블록들에서의 참조 샘플들에 관한 공간적 예측을 이용하여 인코딩된다. 픽처의 인터 코딩식 (inter-coded; P 또는 B) 슬라이스에서의 비디오 블록들은 동일한 픽처의 이웃 블록들에서의 참조 샘플들에 관한 공간적 예측 또는 다른 참조 픽처들에서의 참조 샘플들에 관한 시간적 예측을 이용할 수도 있다. 픽처들은 프레임들이라고 지칭될 수도 있고, 참조 픽처들은 참조 프레임들이라고 지칭될 수도 있다.

[0005]

공간적 또는 시간적 예측은 코딩될 블록에 대한 예측 블록이 생겨나게 한다. 잔차 데이터는 코딩될 원래의 블록과 예측 블록 사이의 화소 차이들을 나타낸다. 인터 코딩식 블록이 예측 블록을 형성하는 참조 샘플들의 블록을 가리키는 모션 벡터 (motion vector) 에 따라 인코딩되고, 잔차 데이터는 코딩된 블록 및 예측 블록 사이의 차이를 나타낸다. 인트라 코딩식 블록이 인트라 코딩 모드 및 잔차 데이터에 따라 인코딩된다. 추가 압축을 위해, 잔차 데이터는 화소 도메인으로부터 변환 도메인으로 변환될 수도 있으며, 결과적으로 잔차

계수들이 생겨나며, 그 계수들은 그 다음에 양자화될 수도 있다. 처음에는 2차원 어레이로 배열된 양자화된 계수들은, 계수들의 1차원 벡터를 생성하기 위하여 스캔될 수도 있고, 엔트로피 코딩이 더 많은 압축을 달성하기 위해 적용될 수도 있다.

발명의 내용

[0006]

본 개시물은 비디오 인코딩 및 디코딩 기법들에 관련된다. 특히, 본 개시물은 팔레트 (palette) -기반 코딩 모드를 이용하여 비디오 데이터를 인코딩 및 디코딩하는 기법들을 설명한다. 팔레트-기반 코딩 모드에서, 비디오 데이터의 블록에 대한 화소 값들은 비디오 데이터의 블록에 연관된 컬러 값들의 팔레트를 기준으로 코딩될 수도 있다. 컬러 값들의 팔레트는 비디오 인코더에 의해 결정될 수도 있고 특정 블록에 대해 가장 일반적인 컬러 값들을 포함할 수도 있다. 비디오 인코더는 컬러 값들의 팔레트로의 인덱스를 비디오 데이터의 블록에서의 각각의 화소에 배정하고, 그러한 인덱스를 비디오 디코더로 시그널링할 수도 있다. 비디오 디코더는 그러면 무슨 컬러 값을 블록에서의 특정 화소에 대해 사용할지를 팔레트로의 인덱스를 사용하여 결정할 수도 있다.

[0007]

팔레트에서의 인덱스들을 시그널링하는 것 외에도, 비디오 인코더가 팔레트 자체를 인코딩된 비디오 비트스트림에서 또한 송신할 수도 있다. 팔레트를 송신하기 위한 기법들은 팔레트 값들을 명시적으로 시그널링하는 것, 그리고 하나 이상의 이전에 코딩된 블록들로부터의 팔레트 엔트리들로부터 현재 블록에 대한 팔레트 엔트리들을 예측하는 것을 포함할 수도 있다. 본 개시물은 비디오 데이터의 이전에 코딩된 블록들로부터의 어떤 팔레트 엔트리들이 비디오 데이터의 현재 블록에 재사용될 수도 있는지를 나타내는 하나 이상의 선택스 엘리먼트들 (예컨대, 이진 예측 벡터) 을 시그널링하는 기법들을 설명한다. 본 개시물은 또한 이진 예측 벡터를 인코딩 및 디코딩하는 기법들을 설명한다.

[0008]

본 개시물의 하나의 예에서, 비디오를 디코딩하는 방법은 비디오 데이터의 현재 블록에 대한 인코딩된 이진 예측 벡터를 수신하는 단계, 런-길이 디코딩 기법을 사용하여 인코딩된 이진 예측 벡터를 디코딩하는 단계, 이진 예측 벡터에 기초하여 비디오 데이터의 현재 블록에 대한 팔레트를 생성하는 단계로서, 이진 예측 벡터는 이전에 사용된 팔레트 엔트리들이 비디오 데이터의 현재 블록에 대한 팔레트에 재사용되는지의 여부를 나타내는 엔트리들을 포함하는, 상기 비디오 데이터의 현재 블록에 대한 팔레트를 생성하는 단계, 및 그 팔레트를 사용하여 비디오 데이터의 현재 블록을 디코딩하는 단계를 포함한다.

[0009]

본 개시물의 다른 예에서, 비디오 데이터를 인코딩하는 방법은 비디오 데이터의 현재 블록에 대한 팔레트를 생성하는 단계, 비디오 데이터의 현재 블록에 대한 팔레트에 대한 이진 예측 벡터를 생성하는 단계로서, 이진 예측 벡터는 이전에 사용된 팔레트 엔트리들이 비디오 데이터의 현재 블록에 대한 팔레트에 재사용되는지의 여부를 나타내는 엔트리들을 포함하는, 상기 비디오 데이터의 현재 블록에 대한 팔레트에 대한 이진 예측 벡터를 생성하는 단계, 런-길이 인코딩 기법을 사용하여 이진 예측 벡터를 인코딩하는 단계, 및 그 팔레트를 사용하여 비디오 데이터의 현재 블록을 인코딩하는 단계를 포함한다.

[0010]

본 개시물의 다른 예에서, 비디오 데이터를 디코딩하도록 구성되는 장치가, 비디오 데이터를 저장하도록 구성되는 메모리와, 메모리와 통신하는 비디오 디코더를 포함하고 상기 비디오 디코더는 비디오 데이터의 현재 블록에 대한 인코딩된 이진 예측 벡터를 수신하고, 런-길이 디코딩 기법을 사용하여 인코딩된 이진 예측 벡터를 디코딩하고, 이진 예측 벡터에 기초하여 비디오 데이터의 현재 블록에 대한 팔레트를 생성하는 것으로서, 이진 예측 벡터는 이전에 사용된 팔레트 엔트리들이 비디오 데이터의 현재 블록에 대한 팔레트에 재사용되는지의 여부를 나타내는 엔트리들을 포함하는, 상기 비디오 데이터의 현재 블록에 대한 팔레트를 생성하고, 그리고 그 팔레트를 사용하여 비디오 데이터의 현재 블록을 디코딩하도록 구성된다.

[0011]

본 개시물의 다른 예에서, 비디오 데이터를 인코딩하도록 구성되는 장치는, 비디오 데이터를 저장하도록 구성되는 메모리와, 메모리와 통신하는 비디오 인코더를 포함하고 상기 비디오 인코더는 비디오 데이터의 현재 블록에 대한 팔레트를 생성하고, 비디오 데이터의 현재 블록에 대한 팔레트에 대한 이진 예측 벡터를 생성하는 것으로서, 이진 예측 벡터는 이전에 사용된 팔레트 엔트리들이 비디오 데이터의 현재 블록에 대한 팔레트에 재사용되는지의 여부를 나타내는 엔트리들을 포함하는, 상기 비디오 데이터의 현재 블록에 대한 팔레트에 대한 이진 예측 벡터를 생성하고, 런-길이 인코딩 기법을 사용하여 이진 예측 벡터를 인코딩하고, 그리고 그 팔레트를 사용하여 비디오 데이터의 현재 블록을 인코딩하도록 구성된다.

[0012]

본 개시물의 다른 예에서, 비디오를 디코딩하는 장치는 비디오 데이터의 현재 블록에 대한 인코딩된 이진 예측 벡터를 수신하는 수단, 런-길이 디코딩 기법을 사용하여 인코딩된 이진 예측 벡터를 디코딩하는 수단, 이진 예

측 벡터에 기초하여 비디오 데이터의 현재 블록에 대한 팔레트를 생성하는 수단으로서, 이진 예측 벡터는 이전에 사용된 팔레트 엔트리들이 비디오 데이터의 현재 블록에 대한 팔레트에 재사용되는지의 여부를 나타내는 엔트리들을 포함하는, 상기 비디오 데이터의 현재 블록에 대한 팔레트를 생성하는 수단, 및 그 팔레트를 사용하여 비디오 데이터의 현재 블록을 디코딩하는 수단을 포함한다.

[0013] 본 개시물의 다른 예에서, 비디오 데이터를 인코딩하도록 구성되는 장치가, 비디오 데이터의 현재 블록에 대한 팔레트를 생성하는 수단, 비디오 데이터의 현재 블록에 대한 팔레트에 대한 이진 예측 벡터를 생성하는 수단으로서, 이진 예측 벡터는 이전에 사용된 팔레트 엔트리들이 비디오 데이터의 현재 블록에 대한 팔레트에 재사용되는지의 여부를 나타내는 엔트리들을 포함하는, 상기 비디오 데이터의 현재 블록에 대한 팔레트에 대한 이진 예측 벡터를 생성하는 수단, 런-길이 인코딩 기법을 사용하여 이진 예측 벡터를 인코딩하는 수단, 및 그 팔레트를 사용하여 비디오 데이터의 현재 블록을 인코딩하는 수단을 포함한다.

[0014] 다른 예에서, 본 개시물은 명령들을 저장하는 컴퓨터 관독가능 저장 매체를 설명하며, 그 명령들은, 실행되는 경우, 비디오 데이터를 디코딩하도록 구성되는 디바이스의 하나 이상의 프로세서들로 하여금, 비디오 데이터의 현재 블록에 대한 인코딩된 이진 예측 벡터를 수신하게 하고, 런-길이 디코딩 기법을 사용하여 인코딩된 이진 예측 벡터를 디코딩하게 하고, 이진 예측 벡터에 기초하여 비디오 데이터의 현재 블록에 대한 팔레트를 생성하게 하는 것으로서, 이진 예측 벡터는 이전에 사용된 팔레트 엔트리들이 비디오 데이터의 현재 블록에 대한 팔레트에 재사용되는지의 여부를 나타내는 엔트리들을 포함하는, 비디오 데이터의 현재 블록에 대한 팔레트를 생성하게 하고, 그 팔레트를 사용하여 비디오 데이터의 현재 블록을 디코딩하게 한다.

[0015] 다른 예에서, 본 개시물은 명령들을 저장하는 컴퓨터 관독가능 저장 매체를 설명하며, 그 명령들은, 실행되는 경우, 비디오 데이터를 인코딩하도록 구성되는 디바이스의 하나 이상의 프로세서들로 하여금, 비디오 데이터의 현재 블록에 대한 팔레트를 생성하게 하고, 비디오 데이터의 현재 블록에 대한 팔레트에 대한 이진 예측 벡터를 생성하게 하는 것으로서, 이진 예측 벡터는 이전에 사용된 팔레트 엔트리들이 비디오 데이터의 현재 블록에 대한 팔레트에 재사용되는지의 여부를 나타내는 팔레트들을 포함하는, 상기 비디오 데이터의 현재 블록에 대한 팔레트에 대한 이진 예측 벡터를 생성하게 하고, 런-길이 인코딩 기법을 사용하여 이진 예측 벡터를 인코딩하게 하고, 그 팔레트를 사용하여 비디오 데이터의 현재 블록을 인코딩하게 한다.

[0016] 하나 이상의 예들의 세부사항들은 첨부 도면들 및 다음의 설명에서 언급된다. 다른 특징들, 목적들, 및 이 점들은 상세한 설명 및 도면들로부터, 그리고 청구항들로부터 명확하게 될 것이다.

도면의 간단한 설명

[0017] 도 1은 본 개시물에서 설명되는 기법들을 구현할 수도 있는 일 예의 비디오 코딩 시스템을 도시하는 블록도이다.

도 2는 본 개시물에서 설명되는 기법들을 구현할 수도 있는 일 예의 비디오 인코더를 도시하는 블록도이다.

도 3은 본 개시물에서 설명되는 기법들을 구현할 수도 있는 일 예의 비디오 디코더를 도시하는 블록도이다.

도 4는 도 2의 비디오 인코더의 일 예 팔레트 기반 인코딩 부를 도시하는 블록도이다.

도 5는 본 개시물의 기법들에 따른 일 예의 팔레트 예측 기법을 도시하는 개념도이다.

도 6은 본 개시물의 기법들에 따른 일 예의 이진 예측 벡터 인코딩 기법을 도시하는 개념도이다.

도 7은 도 3의 비디오 인코더의 일 예의 팔레트 기반 디코딩 부를 도시하는 블록도이다.

도 8은 본 개시물의 일 예의 인코딩 방법을 도시하는 흐름도이다.

도 9는 본 개시물의 일 예의 디코딩 방법을 도시하는 흐름도이다.

도 10은 본 개시물의 기법들과 일치하게, 팔레트 기반 비디오 코딩을 위한 팔레트 엔트리들을 결정하는 일 예를 도시하는 개념도이다.

발명을 실시하기 위한 구체적인 내용

[0018] 본 개시물은 비디오 코딩과 압축을 위한 기법들을 포함한다. 특히, 본 개시물은 비디오 데이터의 팔레트 기반 코딩을 위한 기법들을 설명한다. 전통적인 비디오 코딩에서, 이미지들은 연속 톤이고 공간적으로 매끄러운 것으로 가정된다. 이를 가정들에 기초하여, 블록 기반 변환, 필터링 등과 같은 다양한 도구들이 개발되

었고, 이러한 도구들은 자연 콘텐츠 비디오들에 대해 양호한 성능을 보여 주었다.

[0019] 그러나, 원격 테스크톱, 협력 작업 및 무선 디스플레이 같은 애플리케이션들에서, 컴퓨터 생성된 스크린 콘텐츠(예컨대, 이를테면 텍스트 또는 컴퓨터 그래픽)는 압축될 지배적인 콘텐츠일 수도 있다. 이 유형의 콘텐츠는 이산 톤을 갖고 날카로운 라인(sharp line)들과, 높은 콘트라스트 물체 경계들을 특징으로 하는 경향이 있다. 연속-톤 및 평활도의 가정은 스크린 콘텐츠에 더 이상 적용할 수도 없고, 따라서 전통적인 비디오 코딩 기법들은 스크린 콘텐츠를 포함하는 비디오 데이터를 압축하기 위한 효율적인 방도들이 아닐 수도 있다.

[0020] 팔레트 기반 코딩 기법들이 스크린 생성된 콘텐츠 코딩에 특히 적합할 수도 있다. 예를 들어, 비디오 데이터의 특정 영역이 비교적 적은 수의 컬러들을 갖는다고 가정한다. 비디오 코더(예컨대, 비디오 인코더 또는 비디오 디코더)가 특정 영역(예컨대, 주어진 블록)의 비디오 데이터를 표현하기 위한 컬러들의 테이블로서 이른바 "팔레트"를 형성할 수도 있다. 각각의 화소는 화소의 컬러를 표현하는 팔레트에서의 엔트리와 연관될 수도 있다. 예를 들어, 비디오 코더는 화소 값을 팔레트에서의 적절한 값에 관련시키는 인덱스를 코딩 할 수도 있다.

[0021] 비디오 데이터의 팔레트 기반 코딩을 위한 기법들은 하나 이상의 다른 코딩 기법들, 이를테면 인터 예측 또는 인트라 예측 코딩을 위한 기법들과 함께 사용될 수도 있다. 예를 들어, 아래에서 더욱 상세히 설명되는 바와 같이, 인코더 또는 디코더, 또는 결합된 인코더-디코더(코덱)가, 인터 예측 및 인트라 예측 코딩, 그리고 팔레트 기반 코딩을 수행하도록 구성될 수도 있다.

[0022] 일부 예들에서, 팔레트 기반 코딩 기법들은 하나 이상의 비디오 코딩 표준들과 함께 사용하기 위해 구성될 수도 있다. 일부 예의 비디오 코딩 표준들은 ITU-T H.261, ISO/IEC MPEG-1 비주얼, ITU-T H.262 또는 ISO/IEC MPEG-2 비주얼, ITU-T H.263, ISO/IEC MPEG-4 비주얼 그리고 스케일러블 비디오 코딩(Scalable Video Coding, SVC) 및 멀티뷰 비디오 코딩(Multiview Video Coding, MVC) 확장본들을 포함한 ITU-T H.264(또한 ISO/IEC MPEG-4 AVC로 알려짐)를 포함한다. 다른 예에서, 팔레트 기반 코딩 기법들은 고 효율 비디오 코딩(HEVC)과 함께 사용하기 위해 구성될 수도 있다. HEVC는, ITU-T 비디오 코딩 전문가 그룹(Video Coding Experts Group, VCEG) 및 ISO/IEC 등 화상 전문가 그룹(Motion Picture Experts Group, MPEG)의 JCT-VC(Joint Collaboration Team on Video Coding)에 의해 개발된 새로운 비디오 코딩 표준이다.

[0023] 최근에, HEVC의 설계가 ITU-T 비디오 코딩 전문가 그룹(VCEG) 및 ISO/IEC 등 화상 전문가 그룹(MPEG)의 JCT-VC(Joint Collaboration Team on Video Coding)에 의해 완결되었다. 이하 HEVC 버전 1이라고 지칭되는 최신 HEVC 규격은, 2015년 3월 24일 현재 <http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=11885&lang=en>에서 입수 가능한 "ITU-T H.265 (V1)"에 기재되어 있다. 문서 『ITU-T H.265, SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS, Infrastructure of Audiovisual Services - Coding of Moving Video, "High Efficiency Video Coding," April 2013』이 HEVC 표준을 또한 설명한다. 이후로는 RExt라고 지칭되는 최근의 범위 확장 규격이, 2015년 3월 24일 현재 <http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=12296&lang=en>에 입수 가능한 "ITU-T H.265 (V2)"에 기재되어 있다.

[0024] 예의 팔레트 기반 코딩 기법들을 사용하는 경우, 비디오 인코더가, 블록에 대한 팔레트를 결정하며(예컨대, 팔레트를 명시적으로 코딩하며, 그것을 예측하며, 또는 그것들의 조합을 행하며), 각각의 화소의 값을 표현하기 위해 팔레트에서의 엔트리의 위치를 찾아내고, 화소 값을 팔레트에 관련시키는 화소들에 대한 인덱스 값들과 함께 블록을 인코딩함으로써 비디오 데이터의 블록을 인코딩할 수도 있다. 비디오 디코더가, 인코딩된 비트스트림으로부터, 블록에 대한 팔레트, 그리고 블록의 화소들에 대한 인덱스 값을 획득할 수도 있다. 비디오 디코더는 블록의 화소 값을 복원하기 위해 화소들의 인덱스 값을 팔레트의 엔트리들에 관련시킬 수도 있다. 위의 예는 팔레트 기반 코딩의 일반적인 설명을 제공하기 위한 것이다.

[0025] 팔레트로의 인덱스 값을 시그널링하는 것 외에도, 비디오 인코더가 팔레트 자체를 인코딩된 비디오 비트스트림에서 또한 송신할 수도 있다. 팔레트를 송신하기 위한 기법들은 팔레트 값을 명시적으로 시그널링하는 것, 그리고 하나 이상의 이전에 코딩된 블록들로부터의 팔레트 엔트리들로부터 현재 블록에 대한 팔레트 엔트리들을 예측하는 것을 포함할 수도 있다. 팔레트 예측 기법들은 이전 예측 벡터에서의 각각의 선택스 엘리먼트의 값이 비디오 데이터의 이전에 코딩된 블록으로부터의 팔레트 엔트리가 비디오 데이터의 현재 블록에 재사용될 수도 있는지의 여부를 나타내는 선택스 엘리먼트들의 문자열(예컨대, 이전 예측 벡터)을 시그널링하는 것을 포함할 수도 있다. 이러한 이전 예측 벡터들은 큰 팔레트 사이즈들의 경우 및/또는 다수의 팔레트 엔트리들이 팔레트 예측 프로세스에서의 가능한 사용을 위해 저장되는 경우에 길 수도 있다. 큰 이전 예측 벡터

터들은 용인될 수 없는 비트 레이트 레벨들을 야기할 수도 있다. 이들 단점들의 측면에서, 본 개시물은 팔레트 예측 프로세스에서 이진 예측 벡터들을 인코딩 및 디코딩하는 기법들을 설명한다.

[0026] 도 1은 본 개시물의 기법들을 이용할 수도 있는 일 예의 비디오 코딩 시스템 (10) 을 도시하는 블록도이다. 본원에서 사용되는 바와 같이, "비디오 코더"라는 용어는 비디오 인코더들 및 비디오 디코더들 양쪽 모두를 일반적으로 지칭한다. 본 개시물에서, "비디오 코딩" 또는 "코딩"이란 용어들은 비디오 인코딩 또는 비디오 디코딩 중 어느 하나를 일반적으로 지칭할 수도 있다. 비디오 코딩 시스템 (10) 의 비디오 인코더 (20) 및 비디오 디코더 (30) 는 본 개시물에서 설명되는 다양한 예들에 따라 팔레트 기반 비디오 코딩을 위한 기법들을 수행하도록 구성될 수도 있는 디바이스들의 예들을 나타낸다. 예를 들어, 비디오 인코더 (20) 와 비디오 디코더 (30) 는 팔레트 기반 코딩 또는 비-팔레트 기반 코딩 중 어느 하나를 사용하여 비디오 데이터의 다양한 블록들, 이를테면 고 효율 비디오 코딩 (HEVC) 표준에서와 같은 CU들 또는 PU들을, 선택적으로 코딩하도록 구성될 수도 있다. 비-팔레트 기반 코딩 모드들은 다양한 인터 예측 시간적 코딩 모드들 또는 인터 예측 공간적 코딩 모드들, 이를테면 HEVC 버전 1에 의해 명시된 다양한 코딩 모드들을 지칭할 수도 있다.

[0027] 도 1에 도시된 바와 같이, 비디오 코딩 시스템 (10) 은 소스 디바이스 (12) 와 목적지 디바이스 (14) 를 구비한다. 소스 디바이스 (12) 는 인코딩된 비디오 데이터를 생성한다. 따라서, 소스 디바이스 (12) 는 비디오 인코딩 디바이스 또는 비디오 인코딩 장치라고 지칭될 수도 있다. 목적지 디바이스 (14) 는 소스 디바이스 (12) 에 의해 생성된 인코딩된 비디오 데이터를 디코딩할 수도 있다. 따라서, 목적지 디바이스 (14) 는 비디오 디코딩 디바이스 또는 비디오 디코딩 장치라고 지칭될 수도 있다. 소스 디바이스 (12) 와 목적지 디바이스 (14) 는 비디오 코딩 디바이스들 또는 비디오 코딩 장치들의 예들일 수도 있다. 소스 디바이스 (12), 목적지 디바이스 (14), 또는 둘 다의 다양한 구현예들이, 하나 이상의 프로세서들과 하나 이상의 프로세서들에 커플링된 메모리를 포함할 수도 있다. 메모리는, 본원에서 설명된 바와 같이, RAM, ROM, EEPROM, 플래시 메모리, 또는 명령들 또는 데이터 구조들의 형태의 원하는 프로그램 코드를 저장하는데 사용될 수 있고 컴퓨터에 의해 액세스 가능할 수 있는 임의의 다른 매체를 비제한적으로 포함할 수도 있다.

[0028] 소스 디바이스 (12) 와 목적지 디바이스 (14) 는 데스크톱 컴퓨터들, 모바일 컴퓨팅 디바이스들, 노트북 (예컨대, 랩톱) 컴퓨터들, 태블릿 컴퓨터들, 셋톱 박스들, 이른바 "스마트" 폰들과 같은 전화기 핸드셋들, 텔레비전들, 카메라들, 디스플레이 디바이스들, 디지털 미디어 플레이어들, 비디오 게이밍 콘솔들, 차재 컴퓨터들 등을 포함한 다양한 범위의 디바이스들을 포함할 수도 있다.

[0029] 목적지 디바이스 (14) 는 소스 디바이스 (12) 로부터의 인코딩된 비디오 데이터를 채널 (16) 을 통해 수신할 수도 있다. 채널 (16) 은 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로 인코딩된 비디오 데이터를 이동시킬 수 있는 하나 이상의 매체들 또는 디바이스들을 포함할 수도 있다. 하나의 예에서, 채널 (16) 은 소스 디바이스 (12) 가 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 직접 실시간으로 송신하는 것을 가능하게 하는 하나 이상의 통신 매체들을 포함할 수도 있다. 이 예에서, 소스 디바이스 (12) 는 인코딩된 비디오 데이터를 통신 표준, 이를테면 무선 통신 프로토콜에 따라 변조할 수도 있고, 변조된 비디오 데이터를 목적지 디바이스 (14) 로 송신할 수도 있다. 하나 이상의 통신 매체들은 무선 및/또는 유선 통신 매체들, 이를테면 라디오 주파수 (RF) 스펙트럼 또는 하나 이상의 물리적 송신 라인들을 포함할 수도 있다. 하나 이상의 통신 매체들은 패킷 기반 네트워크, 이를테면 로컬 영역 네트워크, 광역 네트워크, 또는 글로벌 네트워크 (예컨대, 인터넷) 의 일부를 형성할 수도 있다. 하나 이상의 통신 매체들은 라우터들, 스위치들, 기지국들, 또는 소스 디바이스 (12) 로부터 목적지 디바이스 (14) 로의 통신을 용이하게 하는 다른 장비를 포함할 수도 있다.

[0030] 다른 예에서, 채널 (16) 은 소스 디바이스 (12) 에 의해 생성된 인코딩된 비디오 데이터를 저장하는 저장 매체를 포함할 수도 있다. 이 예에서, 목적지 디바이스 (14) 는 예컨대, 디스크 액세스 또는 카드 액세스를 통해 저장 매체에 액세스할 수도 있다. 저장 매체는 블루 레이 디스크들, DVD들, CD-ROM들, 플래시 메모리, 또는 인코딩된 비디오 데이터를 저장하기 위한 다른 적합한 디지털 저장 매체들과 같은 다양한 국부적으로 액세스되는 데이터 저장 매체들을 포함할 수도 있다.

[0031] 추가의 예에서, 채널 (16) 은 소스 디바이스 (12) 에 의해 생성된 인코딩된 비디오 데이터를 저장하는 파일 서버 또는 다른 중간 저장 디바이스를 포함할 수도 있다. 이 예에서, 목적지 디바이스 (14) 는 파일 서버 또는 다른 중간 저장 디바이스에 저장된 인코딩된 비디오 데이터를 스트리밍 또는 다운로드를 통해 액세스할 수도 있다. 파일 서버는 인코딩된 비디오 데이터를 저장하고 그 인코딩된 비디오 데이터를 목적지 디바이스 (14) 로 송신할 수 있는 유형의 서버일 수도 있다. 예의 파일 서버들은 웹 서버들 (예컨대, 웹사이트용), 파일 전송 프로토콜 (file transfer protocol, FTP) 서버들, 네트워크 부속 스토리지 (network attached storage;

NAS) 디바이스들, HTTP 스트리밍 서버들, 및 로컬 디스크 드라이브들을 포함한다.

[0032] 목적지 디바이스 (14)는 표준 데이터 접속, 이를테면 인터넷 접속을 통해, 인코딩된 비디오 데이터에 액세스할 수도 있다. 예의 유형들의 데이터 접속들은 파일 서버 상에 저장된 인코딩된 비디오 데이터에 액세스하기에 적합한 무선 채널들 (예컨대, Wi-Fi 접속들), 유선 접속들 (예컨대, DSL, 케이블 모뎀 등), 또는 양쪽 모두의 조합들을 포함할 수도 있다. 파일 서버로부터의 인코딩된 비디오 데이터의 송신은 스트리밍 송신, 다운로드 송신, 또는 양쪽 모두의 조합일 수도 있다.

[0033] 그러나, 비디오 코딩에서의 팔레트 기반 코딩 모드를 위한 본 개시물의 기법들은, 무선 애플리케이션들 또는 설정들 (settings)로 제한되지 않는다. 그 기법들은, 다양한 멀티미디어 애플리케이션들, 이를테면 OTA (over-the-air) 텔레비전 브로드캐스트들, 케이블 텔레비전 송신들, 위성 텔레비전 송신들, 예컨대, 인터넷을 통한 스트리밍 비디오 송신들을 지원하는 비디오 코딩, 데이터 저장 매체 상의 저장을 위한 비디오 데이터의 인코딩, 데이터 저장 매체 상에 저장된 비디오 데이터의 디코딩, 또는 다른 애플리케이션들에 적용될 수도 있다.

일부 예들에서, 비디오 코딩 시스템 (10)은 비디오 스트리밍, 비디오 플레이백, 비디오 브로드캐스팅, 및/또는 화상 통화와 같은 애플리케이션들을 지원하기 위해 단방향 또는 양방향 비디오 송신을 지원하도록 구성될 수도 있다.

[0034] 도 1에 예시된 비디오 코딩 시스템 (10)은 단지 일 예이고 본 개시물의 기법들은 인코딩 및 디코딩 디바이스들 간에 입의의 데이터 통신을 반드시 포함하지는 않는 비디오 코딩 설정들 (예컨대, 비디오 인코딩 또는 비디오 디코딩)에 적용될 수도 있다. 다른 예들에서, 데이터는 로컬 메모리로부터 취출되며, 네트워크 등을 통해 스트리밍된다. 비디오 인코딩 디바이스가 데이터를 인코딩하여 메모리에 저장할 수도 있으며, 그리고/또는 비디오 디코딩 디바이스가 메모리로부터 데이터를 취출하여 디코딩할 수도 있다. 많은 예들에서, 인코딩 및 디코딩은, 서로 통신하는 것이 아니라, 단순히 데이터를 메모리에 인코딩하거나 및/또는 메모리로부터 데이터를 취출하여 디코딩하는 디바이스들에 의해 수행된다.

[0035] 도 1의 예에서, 소스 디바이스 (12)는 비디오 소스 (18), 비디오 인코더 (20), 및 출력 인터페이스 (22)를 구비한다. 일부 예들에서, 출력 인터페이스 (22)는 변조기/복조기 (모뎀) 및/또는 송신기를 구비할 수도 있다. 비디오 소스 (18)는 비디오 캡처 디바이스, 예컨대, 비디오 카메라, 이전에 캡처된 비디오 데이터를 포함한 비디오 아카이브, 비디오 콘텐츠 제공자로부터 비디오 데이터를 수신하기 위한 비디오 피드 인터페이스, 및/또는 비디오 데이터를 생성하는 컴퓨터 그래픽 시스템, 또는 비디오 데이터의 이러한 소스들의 조합을 포함할 수도 있다.

[0036] 비디오 인코더 (20)는 비디오 소스 (18)로부터의 비디오 데이터를 인코딩할 수도 있다. 일부 예들에서, 소스 디바이스 (12)는 출력 인터페이스 (22)를 통해 목적지 디바이스 (14)로 인코딩된 비디오 데이터를 직접 송신할 수도 있다. 다른 예들에서, 인코딩된 비디오 데이터는 디코딩 및/또는 플레이백을 위한 목적지 디바이스 (14)에 의한 나중의 액세스를 위해 저장 매체 또는 파일 서버 상에 또한 저장될 수도 있다.

[0037] 도 1의 예에서, 목적지 디바이스 (14)는 입력 인터페이스 (28), 비디오 디코더 (30), 및 디스플레이 디바이스 (32)를 구비한다. 일부 예들에서, 입력 인터페이스 (28)는 수신기 및/또는 모뎀을 구비한다. 입력 인터페이스 (28)는 채널 (16)을 통해 인코딩된 비디오 데이터를 수신할 수도 있다. 디스플레이 디바이스 (32)는 목적지 디바이스 (14)와 통합되거나 또는 그것 외부에 있을 수도 있다. 대체로, 디스플레이 디바이스 (32)는 디코딩된 비디오 데이터를 디스플레이한다. 디스플레이 디바이스 (32)는 액정 디스플레이 (LCD), 플라즈마 디스플레이, 유기 발광 다이오드 (OLED) 디스플레이, 또는 다른 유형의 디스플레이 디바이스와 같은 다양한 디스플레이 디바이스들을 포함할 수도 있다.

[0038] 비디오 인코더 (20)와 비디오 디코더 (30) 각각은 다양한 적합한 회로, 이를테면 하나 이상의 마이크로프로세서들, 디지털 신호 프로세서들 (digital signal processors, DSP들), 주문형 집적회로들 (application-specific integrated circuits, ASIC들), 필드 프로그램가능 게이트 어레이들 (field-programmable gate arrays, FPGA들), 개별 로직, 하드웨어, 또는 그것들의 입의의 조합 중 입의의 것으로서 구현될 수도 있다. 그 기법들이 부분적으로 소프트웨어로 구현되면, 디바이스가 적합한 비일시적 (non-transitory) 컴퓨터 판독 가능 저장 매체 내에 소프트웨어에 대한 명령을 저장할 수도 있고 하나 이상의 프로세서들을 사용하여 하드웨어에서 그 명령들을 실행하여 본 개시물의 기법들을 수행할 수도 있다. 전술한 바 (하드웨어, 소프트웨어, 하드웨어 및 소프트웨어의 조합 등을 포함) 중 입의의 것은 하나 이상의 프로세서들이라고 간주될 수도 있다. 비디오 인코더 (20) 및 비디오 디코더 (30)의 각각은 하나 이상의 인코더들 또는 디코더들 내에 구비될 수도 있고, 그것들 중 어느 하나는 결합형 인코더/디코더 (CODEC)의 일부로서 각각의 디바이스 내에 통합될 수도 있

다.

[0039] 본 개시물은 다른 디바이스, 이를테면 비디오 디코더 (30)에 특정 정보를 "시그널링하는" 또는 "송신하는" 비디오 인코더 (20)에 일반적으로 관련이 있을 수도 있다. "시그널링" 또는 "송신"이란 용어는 압축된 비디오 데이터를 디코딩하는데 사용되는 신택스 엘리먼트들 및/또는 다른 데이터의 통신을 일반적으로 지칭할 수도 있다. 이러한 통신은 실시간 또는 근실시간으로 일어날 수도 있다. 대안으로, 이러한 통신은, 인코딩 시에 신택스 엘리먼트들을 인코딩된 비트스트림으로 컴퓨터 관통가능 저장 매체에 저장하고 그 신택스 엘리먼트들이 이 매체에 저장된 후의 임의의 시간에 디코딩 디바이스에 의해 취출될 수도 있는 경우에 일어날 바와 같이 어떤 기간 (span of time)에 걸쳐 일어날 수도 있다.

[0040] 일부 예들에서, 비디오 인코더 (20)와 비디오 디코더 (30)는 비디오 압축 표준, 이를테면 위에서 언급되고 HEVC 버전 1에서 설명된 HEVC 표준에 따라 동작한다. 기본 HEVC 표준 외에도, HEVC에 대한 스케일러블 비디오 코딩, 멀티뷰 비디오 코딩, 및 3D 코딩 확장본들을 생성하기 위한 진행중인 노력들이 있다. 덧붙여서, 예컨대 본 개시물에서 설명되는 바와 같은, 팔레트 기반 코딩 모드들은, HEVC 표준의 확장본에서 제공될 수도 있다. 일부 예들에서, 팔레트 기반 코딩에 대한 본 개시물에서 설명된 기법들은 다른 비디오 코딩 표준들 또는 장래의 표준들에 따라 동작하도록 구성되는 인코더들 및 디코더들에 적용될 수도 있다. 따라서, HEVC 코덱에서의 코딩 유닛들 (coding units, CU들) 또는 예측 유닛들 (prediction units, PU들)의 코딩을 위한 팔레트 기반 코딩 모드의 애플리케이션이 예의 목적들을 위해 설명된다.

[0041] 다양한 예들에서, 본 개시물에서 설명되는 기법들은 팔레트-기반 코딩에서 팔레트들을 결정, 예측, 및/또는 시그널링하는 다양한 조합들을 위한 기법들을 포함할 수도 있다. 아래에서 더욱 상세히 설명되는 바와 같이, 그 기법들은 팔레트 예측자들을 결정하는 것, 이진 예측 벡터를 생성하는 것, 및 이진 예측 벡터를 인코딩/디코딩하는 것의 임의의 조합을 포함할 수도 있다. 비록 특정 예들이 예시 및 명료함을 목적으로 개별적으로 설명될 수도 있지만, 본 개시물은 본원에서 설명되는 팔레트-기반 코딩을 위한 기법들의 임의의 조합을 생각해 낼 수도 있다.

[0042] 도 4 내지 도 9를 참조하여 아래에서 더 상세히 설명될 바와 같이, 비디오 인코더 (20)는 비디오 데이터의 현재 블록에 대한 팔레트를 생성하고, 비디오 데이터의 현재 블록에 대한 팔레트에 대한 이진 예측 벡터를 생성하는 것으로서, 이진 예측 벡터는 이전에 사용된 팔레트 엔트리들 (즉, 이전에 코딩된 블록들에 대한 팔레트들에서의 엔트리들)이 비디오 데이터의 현재 블록에 대한 팔레트에 재사용되는지의 여부를 나타내는 엔트리들을 포함하는, 상기 비디오 데이터의 현재 블록에 대한 팔레트에 대한 이진 예측 벡터를 생성하고, 런-길이 인코딩 기법을 사용하여 이진 예측 벡터를 인코딩하고, 그 팔레트를 사용하여 비디오 데이터의 현재 블록을 인코딩하도록 구성될 수도 있다. 비슷하게, 비디오 디코더 (30)는 비디오 데이터의 현재 블록에 대한 인코딩된 이진 예측 벡터를 수신하고, 런-길이 디코딩 기법을 사용하여 인코딩된 이진 예측 벡터를 디코딩하고, 이진 예측 벡터에 기초하여 비디오 데이터의 현재 블록에 대한 팔레트를 생성하는 것으로서, 이진 예측 벡터는 이전에 사용된 팔레트 엔트리들이 비디오 데이터의 현재 블록에 대한 팔레트에 재사용되는지의 여부를 나타내는 엔트리들을 포함하는, 상기 비디오 데이터의 현재 블록에 대한 팔레트를 생성하고, 그리고 팔레트를 사용하여 비디오 데이터의 현재 블록을 디코딩하도록 구성될 수도 있다.

[0043] HEVC 및 다른 비디오 코딩 표준들에서, 비디오 시퀀스가 일련의 퍽처들을 통상 포함한다. 퍽처들은 "프레임들"이라고 또한 지칭될 수도 있다. 퍽처가, S_L , S_{Cb} 및 S_{Cr} 로 표시되는 3 개의 샘플 어레이들을 포함할 수도 있다. S_L 은 루마 샘플들의 2차원 어레이 (즉, 블록)이다. S_{Cb} 는 Cb 색차 샘플들의 2차원 어레이이다. S_{Cr} 은 Cr 색차 샘플들의 2차원 어레이이다. 색차 샘플들은 본원에서 "크로마" 샘플들이라고 또한 지칭될 수도 있다. 다른 경우들에서, 퍽처가 모노크롬일 수도 있고 루마 샘플들의 어레이만을 포함할 수도 있다.

[0044] 퍽처의 인코딩된 표현을 생성하기 위해, 비디오 인코더 (20)는 코딩 트리 유닛 (coding tree unit, CTU)들의 세트를 생성할 수도 있다. CTU들의 각각은, 루마 샘플들의 코딩 트리 블록 (coding tree block), 크로마 샘플들의 두 개의 대응 코딩 트리 블록들, 및 코딩 트리 블록들의 샘플들을 코딩하는데 사용된 신택스 구조들일 수도 있다. 코딩 트리 블록이 샘플들의 NxN 블록일 수도 있다. CTU가 "트리 블록" 또는 "최대 코딩 유닛 (largest coding unit; LCU)"이라고 또한 지칭될 수도 있다. HEVC의 CTU들은 다른 표준들, 이를테면 H.264/AVC의 매크로블록들과 대체로 유사할 수도 있다. 그러나, CTU가 특정 사이즈로 반드시 제한되는 것은 아니고 하나 이상의 코딩 유닛들 (CU들)을 포함할 수도 있다. 슬라이스가 레스터 스캔에서 연속하여 순서화된 정수 개의 CTU들을 포함할 수도 있다.

- [0045] 코딩된 CTU를 생성하기 위해, 비디오 인코더 (20)는 CTU의 코딩 트리 블록들에 대해 큐드트리 구획화를 재귀적으로 수행하여 코딩 트리 블록들을 코딩 블록들로 나눌 수도 있으며, 그래서 그 이름이 "코딩 트리 유닛들"이다. 코딩 블록이 샘플들의 NxN 블록이다. CU가, 루마 샘플 어레이, Cb 샘플 어레이 및 Cr 샘플 어레이를 갖는 꽉쳐의 루마 샘플들의 코딩 블록 및 크로마 샘플들의 대응하는 두 개의 코딩 블록들과, 그 코딩 블록들의 샘플들을 코딩하는데 사용된 신팩스 구조들일 수도 있다. 비디오 인코더 (20)는 CU의 코딩 블록을 하나 이상의 예측 블록들로 구획화할 수도 있다. 예측 블록이 동일한 예측이 적용되는 샘플들의 직사각형 (즉, 정사각형 또는 정사각형 아닌) 블록일 수도 있다. CU의 예측 유닛 (PU)이 꽉쳐의 루마 샘플들의 예측 블록, 크로마 샘플들의 두 개의 대응하는 예측 블록들, 및 예측 블록 샘플들을 예측하는데 사용된 신팩스 구조들일 수도 있다. 비디오 인코더 (20)는 CU의 각각의 PU의 루마, Cb 및 Cr 예측 블록들에 대한 예측 루마, Cb 및 Cr 블록들을 생성할 수도 있다.
- [0046] 비디오 인코더 (20)는 PU에 대한 예측 블록들을 생성하기 위해 인트라 예측 또는 인터 예측을 사용할 수도 있다. 비디오 인코더 (20)가 PU의 예측 블록들을 생성하기 위해 인트라 예측을 사용하면, 비디오 인코더 (20)는 그 PU에 연관된 꽉쳐의 디코딩된 샘플들에 기초하여 그 PU의 예측 블록들을 생성할 수도 있다.
- [0047] 비디오 인코더 (20)가 PU의 예측 블록들을 생성하기 위해 인터 예측을 사용하면, 비디오 인코더 (20)는 그 PU에 연관된 꽉쳐 이외의 하나 이상의 꽉쳐들의 디코딩된 샘플들에 기초하여 그 PU의 예측 블록들을 생성할 수도 있다. 비디오 인코더 (20)는 PU의 예측 블록들을 생성하기 위해 단-예측 (uni-prediction) 또는 양-예측 (bi-prediction)을 사용할 수도 있다. 비디오 인코더 (20)가 단-예측을 사용하여 PU에 대한 예측 블록들을 생성하는 경우, 그 PU는 단일 모션 벡터 (MV)를 가질 수도 있다. 비디오 인코더 (20)가 양-예측을 사용하여 PU에 대한 예측 블록들을 생성하는 경우, PU는 두 개의 MV들을 가질 수도 있다.
- [0048] 비디오 인코더 (20)가 CU의 하나 이상의 PU들에 대해 하나 이상의 예측 블록들 (예컨대, 예측 루마, Cb 및 Cr 블록들)을 생성한 후, 비디오 인코더 (20)는 그 CU에 대해 잔차 블록을 생성할 수도 있다. 잔차 블록에서의 각각의 샘플은 CU의 예측 블록들 중 하나의 예측 블록에서의 샘플과 CU의 원래의 코딩 블록에서의 대응 샘플 간의 차이를 나타낼 수도 있다. 예를 들어, 비디오 인코더 (20)는 CU에 대한 루마 잔차 블록을 생성할 수도 있다. CU의 루마 잔차 블록에서의 각각의 샘플은 CU의 예측 루마 블록들 중 하나의 예측 루마 블록에서의 루마 샘플과 CU의 원래의 루마 코딩 블록에서의 대응하는 샘플 간의 차이를 나타낸다. 덧붙여서, 비디오 인코더 (20)는 CU에 대한 Cb 잔차 블록을 생성할 수도 있다. CU의 Cb 잔차 블록에서의 각각의 샘플은 CU의 예측 Cb 블록들 중 하나의 예측 Cb 블록에서의 Cb 샘플과 CU의 원래의 Cb 코딩 블록에서의 대응 샘플 간의 차이를 나타낼 수도 있다. 비디오 인코더 (20)는 CU에 대한 Cr 잔차 블록을 또한 생성할 수도 있다. CU의 Cr 잔차 블록에서의 각각의 샘플은 CU의 예측 Cr 블록들 중 하나의 예측 Cr 블록에서의 Cr 샘플 및 CU의 원래의 Cr 코딩 블록에서의 대응 샘플 간의 차이를 나타낼 수도 있다.
- [0049] 더욱이, 비디오 인코더 (20)는 큐드트리 구획화를 사용하여 CU의 잔차 블록 (예컨대, 루마, Cb 및 Cr 잔차 블록들)을 하나 이상의 변환 블록들 (예컨대, 루마, Cb 및 Cr 변환 블록들)로 분해할 수도 있다. 변환 블록이 동일한 변환이 적용되는 샘플들의 직사각형 블록일 수도 있다. CU의 변환 유닛 (transform unit, PU)이 루마 샘플들의 변환 블록, 크로마 샘플들의 2 개의 대응하는 변환 블록들, 및 변환 블록 샘플들을 변환하는데 사용된 신팩스 구조들일 수도 있다. 따라서, CU의 각각의 TU는 루마 변환 블록, Cb 변환 블록, 및 Cr 변환 블록에 연관될 수도 있다. TU에 연관된 루마 변환 블록은 CU의 루마 잔차 블록의 서브-블록일 수도 있다. Cb 변환 블록은 CU의 Cb 잔차 블록의 서브-블록일 수도 있다. Cr 변환 블록은 CU의 Cr 잔차 블록의 서브-블록일 수도 있다.
- [0050] 비디오 인코더 (20)는 하나 이상의 변환들을 TU의 변환 블록에 적용하여 TU에 대한 계수 블록을 생성할 수도 있다. 계수 블록이 변환 계수들의 2차원 어레이일 수도 있다. 변환 계수가 스칼라 양일 수도 있다. 예를 들어, 비디오 인코더 (20)는 하나 이상의 변환들을 TU의 루마 변환 블록에 적용하여 그 TU에 대한 루마 계수 블록을 생성할 수도 있다. 비디오 인코더 (20)는 하나 이상의 변환들을 TU의 Cb 변환 블록에 적용하여 TU에 대한 Cb 계수 블록을 생성할 수도 있다. 비디오 인코더 (20)는 하나 이상의 변환들을 TU의 Cr 변환 블록에 적용하여 TU에 대한 Cr 계수 블록을 생성할 수도 있다.
- [0051] 계수 블록 (예컨대, 루마 계수 블록, Cb 계수 블록 또는 Cr 계수 블록)을 생성한 후, 비디오 인코더 (20)는 그 계수 블록을 양자화할 수도 있다. 양자화는 변환 계수들이 그 변환 계수들을 표현하는데 사용된 데이터의 양을 가능한 한 줄이도록 양자화되어서, 추가의 압축을 제공하는 프로세스를 일반적으로 지칭한다. 비디오 인코더 (20)가 계수 블록을 양자화한 후, 비디오 인코더 (20)는 양자화된 변환 계수들을 나타내는 신팩스

엘리먼트들을 엔트로피 인코딩할 수도 있다. 예를 들어, 비디오 인코더 (20) 는 양자화된 변환 계수들을 나타내는 십백스 엘리먼트들에 대해 콘텍스트 적응 이진 산술 코딩 (Context-Adaptive Binary Arithmetic Coding; CABAC) 을 수행할 수도 있다. 비디오 인코더 (20) 는 엔트로피 인코딩된 십백스 엘리먼트들을 비트스트림으로 출력할 수도 있다.

[0052] CABAC 인코딩을 십백스 엘리먼트에 적용하기 위해, 비디오 인코더는 십백스 엘리먼트를 이진화하여 "빈 (bin) 들"이라고 지칭되는 일련의 하나 이상의 비트들을 형성할 수도 있다. 덧붙여서, 비디오 인코더는 코딩 콘텍스트를 식별할 수도 있다. 코딩 콘텍스트는 특정 값들을 갖는 빈들을 코딩할 확률들을 식별할 수도 있다.

예를 들면, 코딩 콘텍스트는 0 값의 빈을 코딩할 0.7 확률과 1 값의 빈을 코딩할 0.3 확률을 나타낼 수도 있다. 코딩 콘텍스트를 식별한 후, 비디오 인코더 (20) 는 간격을 하부 서브간격 및 상부 서브간격으로 나눌 수도 있다. 서브간격들 중 하나는 값 0과 연관될 수도 있고 나머지 서브간격은 값 1과 연관될 수도 있다.

서브간격들의 폭들은 식별된 코딩 콘텍스트에 의해 연관된 값들에 대해 나타내어진 확률들에 비례할 수도 있다. 십백스 엘리먼트의 빈이 하부 서브간격에 연관된 값을 가지면, 인코딩된 값은 하부 서브간격의 하부 경계와 동일할 수도 있다. 십백스 엘리먼트의 동일한 빈이 상부 서브간격에 연관된 값을 가지면, 인코딩된 값은 상부 서브간격의 하부 경계와 동일할 수도 있다. 십백스 엘리먼트의 다음의 빈을 인코딩하기 위해, 비디오 인코더는 인코딩된 비트의 값에 연관된 서브간격인 간격으로 이들 단계들을 반복할 수도 있다. 비디오 인코더 (20) 가 다음 빈에 대해 이들 단계들을 반복하는 경우, 비디오 인코더는 식별된 코딩 콘텍스트 및 인코딩된 빈들의 실제 값들에 의해 나타내어진 확률들에 기초한 수정된 확률들을 사용할 수도 있다.

[0053] 비디오 디코더 (30) 가 십백스 엘리먼트에 대해 CABAC 디코딩을 수행하는 경우, 그 비디오 디코더는 코딩 콘텍스트를 식별할 수도 있다. 비디오 디코더는 그 다음에 간격을 하부 서브간격 및 상부 서브간격으로 나눌 수도 있다. 서브간격들 중 하나는 값 0과 연관될 수도 있고 나머지 서브간격은 값 1과 연관될 수도 있다.

서브간격들의 폭들은 식별된 코딩 콘텍스트에 의해 연관된 값들에 대해 나타내어진 확률들에 비례할 수도 있다. 인코딩된 값이 하부 서브간격 내에 있으면, 비디오 디코더는 하부 서브간격에 연관된 값을 갖는 빈을 디코딩할 수도 있다. 인코딩된 값이 상부 서브간격 내에 있으면, 비디오 디코더는 상부 서브간격에 연관된 값을 갖는 빈을 디코딩할 수도 있다. 십백스 엘리먼트의 다음의 빈을 디코딩하기 위해, 비디오 디코더는 인코딩된 값을 포함하는 서브간격인 간격으로 이들 단계들을 반복할 수도 있다. 비디오 디코더가 이들 단계들을 다음의 빈에 대해 반복하는 경우, 비디오 디코더는 식별된 코딩 콘텍스트 및 디코딩된 빈들에 의해 나타내어진 확률들에 기초한 수정된 확률들을 사용할 수도 있다. 비디오 디코더는 그 다음에 빈들을 역이진화 (de-binarize) 하여 십백스 엘리먼트를 복원할 수도 있다.

[0054] 모든 십백스 엘리먼트들에 대해 정규 (regular) CABAC 인코딩을 수행하는 것이 아니라, 비디오 인코더는 바이패스 CABAC 코딩을 사용하여 일부 십백스 엘리먼트들 (예컨대, 빈들) 을 인코딩할 수도 있다. 빈에 대해 정규 CABAC 코딩을 수행하는 것보다는 빈에 대해 바이패스 CABAC 코딩을 수행하는 것이 계산적으로 덜 비쌀 수도 있다. 더욱이, 바이패스 CABAC 코딩을 수행하는 것은 높은 정도의 병렬화 및 스루풋을 허용할 수도 있다. 바이패스 CABAC 코딩을 사용하여 인코딩된 빈들이 "바이패스 빈들"이라고 지칭될 수도 있다. 바이패스 빈들을 함께 그룹화하는 것은 비디오 인코더 및 비디오 디코더의 스루풋을 증가시킬 수도 있다. 바이패스 CABAC 코딩 엔진은 단일 사이클에서 여러 빈들을 코딩할 수도 있는 반면, 정규 CABAC 코딩 엔진은 한 사이클에서 단일 빈만을 코딩 가능할 수도 있다. 바이패스 CABAC 코딩 엔진은 바이패스 CABAC 코딩 엔진이 콘텍스트들을 선택하지 않기 때문에 더 간단할 수도 있고, 양 심볼들 (0 및 1) 에 대해 1/2의 확률을 가정할 수도 있다. 결과적으로, 바이패스 CABAC 코딩에서, 간격들은 직접적으로 절반으로 분할된다.

[0055] 비디오 인코더 (20) 는 엔트로피 인코딩된 십백스 엘리먼트들을 포함하는 비트스트림을 출력할 수도 있다. 그 비트스트림은 코딩된 픽처들 및 연관된 데이터의 표현을 형성하는 비트들의 시퀀스를 포함할 수도 있다. 다르게 말하면, 그 비트스트림은 비디오 데이터의 코딩된 표현을 포함할 수도 있다. 그 비트스트림은 네트워크 추상화 계층 (network abstraction layer, NAL) 유닛들의 시퀀스를 포함할 수도 있다. NAL 유닛들의 각각은 NAL 유닛 헤더를 포함하고 원시 바이트 시퀀스 패이로드 (raw byte sequence payload, RBSP) 를 캡슐화 한다. NAL 유닛 헤더는 NAL 유닛 유형 코드를 나타내는 십백스 엘리먼트를 포함할 수도 있다. NAL 유닛의 NAL 유닛 헤더에 의해 명시된 NAL 유닛 유형 코드는 NAL 유닛의 유형을 나타낸다. RBSP가 NAL 유닛 내에 캡슐화되는 정수의 바이트 수를 포함하는 십백스 구조일 수도 있다. 일부 경우들에서, RBSP가 영 비트들을 포함한다.

[0056] 상이한 유형들의 NAL 유닛들이 상이한 유형들의 RBSP들을 캡슐화할 수도 있다. 예를 들어, 제 1 유형의 NAL 유닛이 픽처 파라미터 세트 (picture parameter set, PPS) 에 대한 RBSP를 캡슐화할 수도 있으며, 제 2 유형의

NAL 유닛이 코딩된 슬라이스에 대한 RBSP를 캡슐화할 수도 있으며, 제 3 유형의 NAL 유닛이 SEI에 대한 RBSP를 캡슐화할 수도 있다는 등등이다. 비디오 코딩 데이터에 대한 RBSP들(파라미터 세트들 및 SEI 메시지들에 대한 RBSP과는 대조적임)을 캡슐화하는 NAL 유닛들은, 비디오 코딩 계층(video coding layer, VCL) NAL 유닛 들이라고 지칭될 수도 있다.

[0057] 비디오 디코더(30)는 비디오 인코더(20)에 의해 생성된 비트스트림을 수신할 수도 있다. 덧붙여서, 비디오 디코더(30)는 그 비트스트림으로부터 신택스 엘리먼트들을 디코딩하기 위해 그 비트스트림을 파싱할 수도 있다. 비디오 디코더(30)는 비트스트림으로부터 디코딩된 신택스 엘리먼트들에 적어도 부분적으로 기초하여 비디오 데이터의 픽처들을 복원할 수도 있다. 비디오 데이터를 복원하는 프로세스는 비디오 인코더(20)에 의해 수행된 프로세스에 일반적으로 역할 수도 있다. 예를 들면, 비디오 디코더(30)는 현재 CU의 PU들에 대한 예측 블록들을 결정하기 위해 그 PU들의 MV들을 사용할 수도 있다. 덧붙여서, 비디오 디코더(30)는 현재 CU의 TU들에 연관된 변환 계수 블록들을 역 양자화할 수도 있다. 비디오 디코더(30)는 현재 CU의 TU들에 연관된 변환 블록들을 복원하기 위해 변환 계수 블록들에 대해 역 변환들을 수행할 수도 있다. 비디오 디코더(30)는 현재 CU의 PU들에 대한 예측 블록들의 샘플들을 현재 CU의 TU들의 변환 블록들의 대응하는 샘플들에 가산함으로써 현재 CU의 코딩 블록들을 복원할 수도 있다. 픽처의 각각의 CU에 대한 코딩 블록들을 복원함으로써, 비디오 디코더(30)는 그 픽처를 복원할 수도 있다.

[0058] 위에서 설명된 바와 같이, 본원에서 설명되는 기법들은 비디오 데이터의 팔레트-기반 코딩에 적용될 수도 있다. 팔레트-기반 코딩은 다음의 기법들 중 하나 이상을 일반적으로 포함할 수도 있다. 팔레트 기반 비디오 코딩으로, 비디오 코더(비디오 인코더(20) 및/또는 비디오 디코더(30))가 특정 영역(예컨대, 주어진 블록)의 비디오 데이터를 표현하기 위한 컬러들의 테이블인 이른바 "팔레트"를 형성할 수도 있다. 각각의 화소는 화소의 컬러를 표현하는 팔레트에서의 엔트리와 연관될 수도 있다. 예를 들어, 비디오 인코더(20) 및/또는 비디오 디코더(30)는 화소 값을 팔레트에서의 적절한 값에 관련시키는 인덱스를 코딩할 수도 있다.

[0059] HEVC 프레임워크에 관해, 일 예로서, 팔레트 기반 코딩 기법들이 코딩 유닛(CU) 모드로서 사용되도록 구성될 수도 있다. 다른 예들에서, 팔레트 기반 코딩 기법들은 HEVC의 프레임워크에서 PU 모드로서 사용되도록 구성될 수도 있다. 따라서, CU 모드의 맥락에서 설명되는 다음의 개시된 프로세스들의 모두가, 부가적으로 또는 대안으로, PU들에 적용될 수도 있다. 그러나, 이들 HEVC 기반 예들은 본원에서 설명되는 팔레트 기반 코딩 기법들의 제약 또는 제한으로서 간주되지 않아야 하는데, 이러한 기법들이 다른 현존의 또는 아직 개발되지 않은 시스템들/표준들과는 상관없이 또는 그러한 시스템들/표준들의 부분으로서 작동하기 위해 적용될 수도 있기 때문이다. 이들 경우들에서, 팔레트 코딩을 위한 유닛은 정사각형 블록들, 직사각형 블록들 또는 심지어 비-직사각형 형상의 지역들일 수 있다.

[0060] 문서 『X. Guo and A. Saxena, "RCE4: Summary report of HEVC Range Extension Core Experiments 4 (RCE4) on palette coding for screen content," JCTVC-P0035, San Jose, US, 9-17 Jan. 2014』는 팔레트 기반 모드들의 두 개의 테스트 결과들을 설명하는데, 그 결과들은 특히 스크린 콘텐츠들에 대해, 상당한 BD-레이트(Bjontegaard Distortion-rate) 감소를 성취하는 것이 보고되었다. 두 가지 방법들이 아래에서 간략히 요약되어 있다.

[0061] 하나의 예의 방법에서는, 예를 들어, 문서 『X. Guo, Y. Lu, and S. Li, "RCE4: Test 1. Major-color-based screen content coding," JCTVC-P0108, San Jose, US, 9-17 Jan. 2014』에서 설명된 바와 같이, 히스토그램 기반 알고리즘이 화소들을 분류하는데 사용된다. 특히, 히스토그램에서의 가장 중요한 N 개의 피크 값들은 코딩을 위한 주요 컬러들로서 선택된다. 주요 컬러에 가까운 화소 값들은 주요 컬러에 대해 양자화될 것이다.

임의의 주요 컬러 세트들에 속하지 않는 다른 화소들은 이스케이프 화소들인데, 그것들은 코딩하기 전에 또한 양자화된다. 무손실 코딩에 대해, 양자화는 사용되지 않는다.

[0062] 분류를 사용하는 것에 의해, 코딩 유닛(CU)의 화소들은 컬러 인덱스들로 변환될 수 있다. 그 후, 주요 컬러 수 및 값들이 코딩된다. 그 다음에, 컬러 인덱스들은 다음과 같이 코딩된다:

[0063] • 각각의 화소 라인에 대해, 플래그가 코딩 모드를 나타내기 위해 시그널링된다. 세 가지 모드들, 즉, 수평 모드, 수직 모드, 및 정상 모드가 있다.

[0064] o 그 모드가 수평 모드이면, 전체 라인은 동일한 컬러 인덱스를 공유한다. 이 경우, 컬러 인덱스는 송신된다.

[0065] o 그 모드가 수직 모드이면, 전체 라인은 상측 라인과 동일하다. 이 경우, 아무 것도 송신되

지 않는다. 현재 라인은 상측 라인의 컬러 인덱스들을 복사한다.

[0066] o 그 모드가 정상 모드이면, 각각의 화소 포지션에 대해 그것이 좌측 및 상측 화소들 중 하나의 화소와 동일한지의 여부를 나타내기 위해 플래그가 송신된다. 아니라면, 인덱스 자체가 송신된다.

[0067] 덧붙여, 화소가 이스케이프 화소이면, 그 화소 값은 송신된다.

[0068] 다른 예의 방법에서는, 예를 들어 문서 『L. Guo, W. Pu, M. Karczewicz, J. Sole, R. Joshi, and F. Zou, "RCE4: Results of Test 2 on Palette Mode for Screen Content Coding," JCTVC-P0198, San Jose, US, 9-17 Jan. 2014』에서 설명된 바와 같이, 팔레트 기반 코딩 모드가 CU 모드로서 포함된다. 제 2의 방법의 인코딩 프로세스는 다음을 포함할 수도 있다:

[0069] • 팔레트의 송신: 엔트리-식 예측 스킁이 현재 팔레트를 좌측 CU (현재 코딩되고 있는 CU에 좌측으로 이웃하는 CU) 의 팔레트에 기초하여 인코딩하는데 사용된다. 그 후, 팔레트의 비-예측된 엔트리들이 송신된다.

[0070] • 화소 값들의 송신: CU에서의 화소들은 다음 세 가지 모드들을 사용하여 레스터 스캔 순서로 인코딩된다:

[0071] o "런 모드 (run mode)": 팔레트 인덱스가 처음 시그널링되며, "palette_run" (M) 이 그 뒤를 따른다. 다음 M 개의 팔레트 인덱스들은 처음 시그널링되는 시그널링된 팔레트 인덱스와 동일하다.

[0072] o "상측 복사 모드": 값 "copy_run" (N) 이 다음 N 개의 팔레트 인덱스들이 그것들의 상측 이웃들과 각각 동일하다는 것을 나타내기 위해 송신된다.

[0073] o "화소 모드": 예측 플래그가 먼저 송신된다. 1과 동일한 플래그 값은 복원된 상단 이웃 화소를 예측자로서 사용하여 예측 잔차가 송신된다는 것을 나타낸다. 이 플래그의 값이 0이면, 화소 값은 예측 없이 송신된다.

[0074] 팔레트는 팔레트 코딩된 블록 (예컨대, CU) 에 대한 비트들의 상대적으로 중요한 부분을 형성할 수도 있다. 따라서, 비디오 코더는 (예컨대, "팔레트의 송신"에 관해 위에서 언급했듯이) 팔레트의 하나 이상의 엔트리들을 이전에 코딩된 팔레트의 하나 이상의 엔트리들에 기초하여 예측할 수도 있다.

[0075] 일부 예들에서, 비디오 코더는 팔레트 엔트리들을 예측하는 경우 팔레트 예측자 리스트를 생성할 수도 있다. 예를 들어, 문서 『C. Gisquet, G. Laroche, and P. Onno, "AhG10: Palette predictor stuffing," JCTVC-Q0063』는 팔레트 예측자들을 결정하는 하나의 프로세스를 개시한다. 일부 예들에서, 비디오 코더는 팔레트 예측자 리스트에서의 각각의 아이템이 현재 코딩되고 있는 블록에 대한 팔레트에서의 하나 이상의 엔트리들을 예측하기 위해 사용되는지 (또는 사용되지 않는지) 를 나타내기 위해 불리언 벡터를 사용할 수도 있다.

[0076] 일부 예들에서, 팔레트 예측자 리스트에서의 아이템들의 모두는 이전에 코딩된 팔레트 (예컨대, 이전에 코딩된 블록으로 코딩된 팔레트)로부터 도출된다. 그러나, 이러한 팔레트들은 현재 CU로부터 공간적으로 떨어져 있을 수도 있는데, 이는 팔레트 상관을 상대적으로 약하게 할 수도 있다. 대체로, 팔레트 예측자 테이블을 확장하는 것이 도움이 될 수도 있다 (예컨대, 더욱 정확한 예측자들을 제공할 수도 있으며, 이는 효율성 증대를 초래할 수도 있다). 그러나, 비교적 큰 팔레트 예측자 테이블을 결정 및 사용하는 것은 상대적으로 더 긴 불리언 벡터를 초래한다.

[0077] 팔레트 코딩의 하나의 예에서, 비디오 인코더 (20) 는 팔레트-기반 코딩 모드가 비디오 프레임의 특정 지역을 위해 사용되는지의 여부를 나타내는 플래그 "PLT_Mode_flag"와 같은 신팩스 엘리먼트를 생성할 수도 있다. 예를 들어, PLT_Mode_flag는 비디오 프레임의 슬라이스 레벨, CU-레벨, PU-레벨, 또는 임의의 다른 레벨에서 생성될 수도 있다. 예를 들어, 비디오 인코더 (20) 는 PLT_Mode_flag를 CU 레벨에서 생성하고 PLT_Mode_flag를 인코딩된 비디오 비트스트림에서 시그널링할 수도 있다. 비디오 디코더 (30) 는 그 다음에 PLT_Mode_flag를 인코딩된 비디오 비트스트림의 디코딩 시에 파싱할 수도 있다. 이 예에서, 이 PLT_Mode_flag의 1과 동일한 값이 현재 CU는 팔레트 모드를 사용하여 인코딩된다는 것을 명시한다. 이 경우, 비디오 디코더 (30) 는 팔레트-기반 코딩 모드를 적용하여 CU를 디코딩할 수도 있다. 일부 예들에서, 신팩스 엘리먼트가 CU에 대한 복수의 상이한 팔레트 모드들 중 하나를 나타낼 수도 있다.

[0078] 이 PLT_Mode_flag의 0과 동일한 값이 현재 CU는 팔레트 모드와는 다른 모드를 사용하여 인코딩된다는 것을 명시 한다. 예를 들어, 다양한 인터-예측, 인트라-예측, 또는 다른 코딩 모드들 중 임의의 것이 사용될 수도 있

다. PLT_Mode_flag의 값이 0인 경우, 추가의 정보가 어떤 특정 모드가 각각의 CU를 인코딩하기 위해 사용되는지를 시그널링하기 위해 송신될 수도 있으며, 이러한 특정 모드는, 보통, HEVC 코딩 모드일 수도 있다. PLT_Mode_flag의 사용은 예의 목적을 위해 설명된다. 다른 예들에서, 그러나, 멀티-비트 코드들과 같은 다른 신택스 엘리먼트들은 팔레트-기반 코딩 모드가 CU (또는 다른 예들에서의 PU)에 대해 사용될 것인지의 여부를 나타내기 위해 또는 복수의 모드들 중 어떤 것이 사용될 것인지를 나타내기 위해 사용될 수도 있다.

[0079] PLT_Mode_flag 또는 다른 신택스 엘리먼트는 더 높은 레벨에서 또한 송신될 수도 있다. 예를 들어, PLT_Mode_flag는 슬라이스 레벨에서 송신될 수도 있다. 이 경우, 그 플래그의 1과 동일한 값이 슬라이스에서의 CU들의 모두가 팔레트 모드를 사용하여 인코딩될 것임을 의미한다 (이는 예컨대 팔레트 모드 또는 다른 모드들에 대한 모드 정보가 CU 레벨에서 송신될 필요가 없다는 것을 의미한다). 마찬가지로, 이 플래그는 꽉 채 파라미터 세트 (PPS), 시퀀스 파라미터 세트 (sequence parameter set, SPS) 또는 비디오 파라미터 세트 (video parameter set, VPS) 레벨에서 시그널링될 수 있다. 또한, 팔레트 모드가 특정 꽉 채, 슬라이스 등에 대해 가능하게 되는지 또는 불가능하게 되는지를 명시하는 플래그가 이들 레벨들 중 하나의 레벨에서 전송될 수 있는 반면, PLT_Mode_flag는 팔레트-기반 코딩 모드가 각각의 CU에 대해 사용되는지의 여부를 나타낸다. 이 경우, 슬라이스, PPS, SPS 또는 VPS 레벨에서 전송된 플래그 또는 다른 신택스 엘리먼트가 팔레트 코딩 모드가 불가능이 됨을 나타낸다면, 일부 예들에서, 각각의 CU에 대해 PLT_Mode_flag를 시그널링하는 것이 필요하지 않을 수도 있다. 대안으로, 슬라이스, PPS, SPS 또는 VPS 레벨에서 전송된 플래그 또는 다른 신택스 엘리먼트가 팔레트 코딩 모드가 가능하게 됨을 나타낸다면, PLT_Mode_flag는 팔레트-기반 코딩 모드가 각각의 CU에 대해 사용되는지 여부를 나타내기 위해 추가로 시그널링될 수도 있다. 다시, 위에서 언급했듯이, CU의 팔레트-기반 코딩을 나타내는 이들 기법들의 애플리케이션이 PU의 팔레트-기반 코딩을 나타내기 위해 부가적으로 또는 대안으로 사용될 수 있다.

[0080] 플래그, 이를테면 PLT_Mode_flag가, 또한 또는 대안적으로 또는 조건부로 송신 또는 유추될 수도 있다. PLT_Mode_flag를 송신하기 위한 또는 그 플래그를 유추하기 위한 조건들은 예들로서, CU의 사이즈, 프레임 유형, 컬러 공간, 컬러 성분, 프레임 사이즈, 프레임 레이트, 스케일러블 비디오 코딩에서의 계층 id 또는 멀티-뷰 코딩에서의 뷰 id 중 하나 이상일 수 있다.

[0081] 팔레트의 생성 및 송신을 위한 기법들이 이제 논의될 것이다. 비디오 인코더 (20)는 비디오 프레임의 특정 레벨 (예컨대, CU)을 인코딩하기 위해 비디오 인코더 (20)에 의해 사용되는 팔레트를 구축 및/또는 복원하기 위해 비디오 디코더 (30)에 의해 사용될 수도 있는 하나 이상의 신택스 엘리먼트들 및 값들을 생성 및 시그널링하도록 구성될 수도 있다. 일부 예들에서, 비디오 인코더 (20)는 각각의 CU에 대한 팔레트를 나타내거나 또는 그렇지 않으면 시그널링할 수도 있다. 다른 예들에서, 비디오 인코더 (20)는 여러 CU들 중에서 공유될 수도 있는 팔레트를 나타내거나 또는 그렇지 않으면 시그널링할 수도 있다.

[0082] 예컨대 포함되는 화소 값들의 수의 측면에서, 팔레트의 사이즈는 고정된 값일 수 있거나 또는 비디오 인코더 (20)에 의해 인코딩된 비디오 비트스트림에서 시그널링될 수 있다. 비디오 디코더 (30)는 인코딩된 비디오 비트스트림으로부터 팔레트 사이즈의 표시를 수신 및 디코딩할 수도 있다. 시그널링은 상이한 성분들에 대해 별개일 수 있거나 또는 단일 사이즈가 모든 성분들에 대해 시그널링될 수 있다. 상이한 성분들은, 예를 들어, 루마 및 크로마 성분들일 수도 있다. 시그널링은 단항 코드들 또는 (예컨대, 팔레트 사이즈의 최대 한도에서 절단되는) 절단된 (truncated) 단항 코드들을 사용할 수 있다. 지수-골롬 (Exponential-Golomb) 또는 라이스-골롬 (Rice-Golomb) 코드들이 또한 사용될 수 있다. 일부 예들에서, 사이즈의 시그널링은 다음의 방도로 행해질 수 있는데, 즉, 팔레트에서의 엔트리를 시그널링한 후, "정지 (stop)" 플래그가 시그널링된다. 이 플래그의 1과 동일한 값이 현재 엔트리가 팔레트에서의 마지막 엔트리임을 명시하며; 이 플래그의 0과 동일한 값이 더 많은 엔트리들이 팔레트에 있다는 것을 명시한다. "정지" 플래그는 이미 구축된 팔레트가 팔레트 사이즈의 최대 한도에 달한다면 인코더에 의해 송신되지 않을 수도 있다. 일부 예들에서, 팔레트의 사이즈는 "플래그 PLT_Mode_flag의 송신"에 대해 위에서 설명된 바와 동일한 방도로 사이드 정보에 기초하여 또한 조건부로 송신되거나 또는 유추될 수 있다.

[0083] 팔레트는 CU에서의 각각의 컬러 성분에 대해 따로따로 송신될 수 있다. 예를 들어, 이 CU의 Y 성분에 대한 팔레트, 이 CU의 U 성분에 대한 다른 팔레트, 및 이 CU의 V 성분에 대한 또 다른 팔레트가 있을 수도 있다. Y 팔레트에 대해, 엔트리는 이 CU에서의 (가정상) 대표적인 Y 값일 수도 있다. 동일한 것이 U 및 V 성분들에 적용된다. 팔레트는 CU에서의 컬러 성분들의 모두를 위해 송신될 수도 있다는 것이 또한 가능하다. 이 예에서, 팔레트에 있는 i-번째 엔트리는 3중의 (Y_i , U_i , V_i)이다. 이 경우, 팔레트는 성분들의 각각에

대한 값들을 포함한다.

[0084] 팔레트의 예측은 위에서 설명된 "팔레트의 송신"에 대한 대안적 접근법이다. 일부 예들에서, 팔레트 예측 기법들이 팔레트 시그널링 기법들과 함께 사용될 수도 있다. 다시 말하면, 비디오 인코더 (20)는 전체 수의 팔레트 엔트리들의 일 부분을 예측하기 위해 비디오 디코더 (30)에 의해 사용될 수도 있는 선택스 엘리먼트들을 시그널링하도록 구성될 수도 있다. 덧붙여서, 비디오 인코더 (20)는 팔레트 엔트리들의 다른 부분을 명시적으로 시그널링하도록 구성될 수도 있다.

[0085] 팔레트 예측 접근법의 하나의 예에서, 각각의 CU에 대해, 하나의 플래그 "pred_pallette_flag"가 송신된다. 이 플래그의 1과 동일한 값이, 현재 CU에 대한 팔레트는 과거의 데이터로부터 예측될 것이고 따라서 팔레트가 송신될 필요가 없다는 것을 명시한다. 이 플래그의 0과 동일한 값은 현재 CU의 팔레트는 송신될 필요가 있다는 것을 의미한다. 그 플래그는 상이한 컬러 성분들에 대해 별개일 수 있거나 (예컨대, 그래서 3 개의 플래그들이 YUV 비디오에서의 CU에 대해 송신될 것이 필요하거나), 또는 단일 플래그가 모든 컬러 성분들에 대해 송신될 수 있다. 예를 들어, 단일 플래그가, 팔레트들이 성분들의 모두에 대해 송신되는지의 여부 또는 성분들의 모두에 대한 팔레트들이 예측될지의 여부를 나타낼 수도 있다.

[0086] 일부 예들에서, 예측은 다음 방식으로 수행될 수 있다. 예측 플래그 값이 1과 동일하면, 현재 CU에 대해, 비디오 인코더 (20)는 하나 이상의 이미 인코딩된 이웃 CU들의 팔레트를 복사한다. 이미 인코딩된 이웃 CU들의 팔레트는 송신 또는 예측되었을 수도 있다. 예를 들어, 복사된 이웃 CU는 좌측 이웃 CU일 수 있다. 좌측 CU의 팔레트는 이용 가능하지 않은 경우 (좌측 CU는 팔레트 모드를 사용하여 인코딩되지 않거나 또는 현재 CU는 꽉치의 첫 번째 열에 있는 경우에서처럼), 팔레트의 사본은 현재 CU 상측 CU로부터일 수 있다. 복사된 팔레트는 또한 다수의 이웃 CU들의 팔레트들의 조합일 수 있다. 예를 들어, 하나 이상의 콘식들, 함수들, 규칙들 등이 복수의 이웃 CU들 중 하나 또는 그것들의 조합의 팔레트들에 기초하여 팔레트를 생성하기 위해 적용될 수도 있다.

[0087] 후보 리스트가 구축될 수도 있고 현재 CU가 팔레트를 복사해 오는 후보 CU를 나타내기 위해 비디오 인코더 (20)에 의해 인덱스가 송신된다는 것이 또한 가능하다. 비디오 디코더 (30)는 동일한 후보 리스트를 구축한 다음 그 인덱스를 사용하여 현재 CU와 함께 사용하기 위한 대응하는 CU의 팔레트를 선택할 수도 있다. 예를 들어, 후보 리스트는 슬라이스 또는 꽉쳐 내의 코딩될 현재 CU를 기준으로 상측의 하나의 CU와 좌측의 하나의 CU를 포함할 수도 있다. 이 예에서, 플래그 또는 다른 선택스 엘리먼트가 후보 선택을 나타내기 위해 시그널링될 수도 있다. 예를 들어, 0과 동일한 송신된 플래그가 사본이 좌측 CU로부터 유래함을 의미하고, 1과 동일한 송신된 플래그가 사본이 상단 CU로부터 유래한다는 것을 의미한다. 비디오 디코더 (30)는 대응하는 이웃 CU로부터 복사될 팔레트를 선택하고 그것을 현재 CU를 디코딩함에 있어서의 사용을 위해 복사한다. 예측은 현재 CU의 인과적 이웃들에서의 가장 빈번한 샘플 값들을 사용하여 또한 도출될 수 있다.

[0088] 팔레트들의 예측은 또한 엔트리-식일 수 있다. 팔레트에서의 각각의 엔트리에 대해, 비디오 인코더 (20)는 플래그를 생성 및 시그널링한다. 주어진 엔트리에 대한 플래그의 1과 동일한 값이 예측된 값 (예를 들어, 좌측 CU 같은 선택된 후보 CU로부터의 대응하는 엔트리) 이 엔트리의 값으로서 사용된다는 것을 명시한다. 플래그의 0과 동일한 값이, 이 엔트리는 예측되지 않고 그것의 값은 비디오 인코더 (20)로부터 비디오 디코더 (30)로 송신될, 예컨대, 비디오 디코더 (30)에 의한 디코딩을 위해 비디오 인코더 (20)에 의해 인코딩된 비트스트림에서 시그널링될 것임을 명시한다.

[0089] "pred_pallette_flag"의 값, 현재 CU의 팔레트를 예측하는데 사용되는 팔레트를 갖는 후보 CU 또는 후보들을 구축하기 위한 규칙들은, "플래그 PLT_Mode_flag의 송신"에 대해 위에서 설명된 바와 동일한 방도로 사이드 정보에 기초하여 또한 조건부로 송신 또는 유추될 수 있다.

[0090] 다음으로, 비디오 인코더 (20)는 어떤 각각의 팔레트 엔트리가 CU에서의 각각의 화소와 연관되는지를 나타내는 맵을 생성 및 시그널링할 수도 있다. 그 맵에 있는 i-번째 엔트리는 CU에 있는 i-번째 포지션에 대응한다. i-번째 엔트리의 1과 동일한 값이 CU에 있는 이 i-번째 로케이션에서의 화소 값이 팔레트에서의 값들 중 하나임을 명시하고, 팔레트 인덱스가 추가로 송신되어서 비디오 디코더 (30)는 그 화소 값을 복원할 수 있다 (단지 하나의 엔트리가 팔레트에 있는 경우, 팔레트 인덱스의 송신은 스킵될 수도 있다). i-번째 엔트리의 0과 동일한 값이, CU에 있는 i-번째 포지션에서의 화소 값은 팔레트 내에 있지 않고 따라서 화소 값은 비디오 디코더 (30)로 명시적으로 송신될 것임을 명시한다.

[0091] CU에 있는 하나의 포지션에서의 화소 값이 팔레트에 있는 값이면, CU에 있는 이웃하는 포지션들이 동일한 화소

값을 가질 높은 확률이 있다는 것이 관측된다. 그래서, 포지션에 대한 팔레트 인덱스 (j라고 쓰는데, 이는 화소 값 s 에 대응함) 를 인코딩한 후, 비디오 인코더 (20) 는 스캔이 상이한 화소 값에 도달하기 전에 CU에서의 동일한 화소 값 (s) 의 연속적인 값들의 수를 나타내기 위해 싱택스 엘리먼트 "런"을 송신할 수도 있다. 예를 들어, 바로 다음 것이 s 와는 상이한 값을 갖는다면, 런 = 0이 송신된다. 다음 것이 s 이지만 뒤의 것이 s 가 아니면, 런 = 1이다.

[0092] 런이 송신되지 않는 경우 (예컨대, 암시적 런 도출), 런의 값은 상수, 예를 들어, 4, 8, 16 등일 수도 있거나, 또는 런의 값은 사이드 정보에 또한 의존적일 수도 있다. 예를 들어, 런의 값은 블록 사이즈에 의존할 수도 있으며, 예컨대, 런은 현재 블록의 폭, 또는 현재 블록의 높이, 또는 현재 블록의 반-폭 (또는 절반-높이), 또는 블록의 폭 및 높이의 분수, 또는 블록의 높이/폭의 배수와 동일하다. 런의 값은 QP, 프레임-유형, 컬러 성분, 컬러 포맷 (예컨대, 444, 422, 420) 및/또는 컬러 공간 (예컨대, YUV, RGB) 에 또한 의존적일 수도 있다.

런의 값은 스캔 방향에 또한 의존할 수도 있다. 다른 예들에서, 런의 값은 다른 유형들의 사이드 정보에 의존할 수도 있다. 런의 값은 고 레벨 싱택스 (예컨대, PPS, SPS) 를 사용하여 또한 시그널링될 수도 있다.

[0093] 일부 예들에서, 맵은 송신될 필요가 없을 수도 있다. 런은 특정 로케이션들에서만 시작할 수도 있다. 예를 들어, 런은 단지 각각의 행의 처음, 또는 N 개 행들마다의 처음에서만 시작할 수도 있다. 시작 로케이션은 상이한 스캔 방향들에 대해 상이할 수도 있다. 예를 들어, 수직 스캔이 사용된다면, 런은 단지 열의 처음 또는 N 개 열들마다의 처음에서만 시작할 수도 있다. 시작 로케이션은 사이드 정보에 의존할 수도 있다. 예를 들어, 시작 로케이션은 각각의 행 또는 각각의 열의 중간 지점, 또는 각각의 행/열의 $1/n$, $2/n, \dots, (n-1)/n$ (즉, 분수들) 일 수도 있다. 시작 로케이션은 QP, 프레임-유형, 컬러 성분, 컬러 포맷 (예컨대, 444, 422, 420) 및/또는 컬러 공간 (예컨대, YUV, RGB) 에 또한 의존할 수도 있다. 다른 예들에서, 런의 시작 포지션은 다른 유형들의 사이드 정보에 의존할 수도 있다. 시작 포지션은 고 레벨 싱택스 (예컨대, PPS, SPS 등) 를 사용하여 또한 시그널링될 수 있다.

[0094] 암시적 시작 포지션 도출과 암시적 런 도출은 결합된다는 것이 또한 가능하다. 예를 들어, 런은 두 개의 이웃하는 시작 포지션들 간의 거리와 동일하다. 시작 점이 모든 행의 처음 (즉, 첫 번째 포지션) 인 경우, 런의 길이는 행이다.

[0095] 스캔 방향은 수직 또는 수평일 수도 있다. 플래그가 각각의 CU에 대해 스캔 방향을 나타내기 위해 송신되는 것이 가능하다. 플래그들은 각각의 컴포넌트에 대해 따로따로 송신될 수도 있거나 또는 단일 플래그가 송신될 수도 있고 지시된 스캔 방향은 모든 컬러 성분들에 적용된다. 45 도 또는 135 도 같은 다른 스캔 방향들이 사용되는 것이 또한 가능하다. 스캔 순서는 고정일 수도 있거나 또는 "플래그 PLT_Mode_flag의 송신"을 위해 위에서 설명된 바와 동일한 방도로 사이드 정보에 의존적일 수도 있다.

[0096] 위에서, 팔레트를 송신하는 방법이 설명되어 있다. 위에서 설명된 예들에 대한 대안이 팔레트를 온 더 플라이 (on-the-fly) 구축하는 것이다. 이 경우, CU의 처음에서는, 엔트리가 팔레트 내에 없고, 비디오 인코더 (20) 가 CU에 있는 포지션들에 대한 화소들의 새로운 값들을 시그널링하므로, 이를 값들은 팔레트에 포함된다.

다시 말하면, 비디오 인코더 (20) 는 화소 값들이 CU에 있는 포지션들에 대해 생성 및 송신되는 대로 그 화소 값들을 팔레트에 추가한다. 그 다음에, 동일한 값들을 갖는 CU에 있는 더 나중의 포지션들이, 비디오 인코더 (20) 가 화소 값들을 송신하는 대신, 예컨대 인덱스 값들로, 팔레트에서의 화소 값들을 참조할 수도 있다.

마찬가지로, 비디오 디코더 (30) 가 CU에 있는 포지션에 대한 새로운 화소 값 (예컨대, 인코더에 의해 시그널링됨) 을 수신하는 경우, 그 비디오 디코더는 그 화소 값을 비디오 디코더 (30) 에 의해 구축된 팔레트에 포함시킨다. CU에서의 더 나중의 포지션들이 팔레트에 추가된 화소 값들을 가지는 경우, 비디오 디코더 (30) 는, 예컨대, CU에서의 화소 값들의 복원을 위해 팔레트에서의 대응하는 화소 값들을 식별하는 인덱스 값들과 같은 정보를 수신할 수도 있다.

[0097] 예컨대 팔레트가 온 더 플라이 동적으로 구축됨에 따라, 최대 팔레트 사이즈가 도달되면, 인코더와 디코더는 팔레트의 엔트리를 제거하기 위해 동일한 메커니즘을 공유한다. 하나의 방법은 팔레트 (FIFO 대기행렬) 에서의 가장 오래된 엔트리를 제거하는 것이다. 다른 방법은 팔레트에서의 가장 적게 사용되는 엔트리를 제거하는 것이다. 다른 것은 교체될 엔트리를 결정하기 위해 양 방법들 (팔레트에서의 시간과 사용 빈도수) 을 가중시키는 것이다. 하나의 예로서, 화소 값 엔트리가 팔레트로부터 제거되고 화소 값이 팔레트에서의 더 나중의 포지션에서 발생한다면, 인코더는 엔트리를 팔레트에 포함시키는 대신 그 화소 값을 송신할 수도 있다.

덧붙여, 또는 대안으로, 예컨대 인코더와 디코더가 CU에 있는 포지션들을 스캔함에 따라, 이러한 화소 값이 제거된 후에 팔레트에 다시 입력될 수 있는 것이 가능하다.

- [0098] 본 개시물은 또한 초기 팔레트 시그널링과 팔레트의 온 더 플라이 도출을 결합하는 것을 도한 고려한다. 하나의 예에서, 초기 팔레트는 화소들의 코딩으로 업데이트될 것이다. 예를 들어, 초기 팔레트를 송신 시, 비디오 인코더 (20) 는 CU에서의 추가적인 로케이션들의 화소 값들이 스캔됨에 따라 값들을 초기 팔레트에 추가하거나 또는 초기 팔레트에 있는 값들을 변경시킬 수도 있다. 비슷하게, 초기 팔레트를 수신 시, 비디오 디코더 (30) 는 CU에서의 추가적인 로케이션들의 화소 값들이 스캔됨에 따라 값들을 초기 팔레트에 추가하거나 또는 초기 팔레트에 있는 값들을 변경시킬 수도 있다. 마찬가지로, 인코더는 현재 CU가 전체 팔레트의 송신을 사용하는지, 또는 온 더 플라이 팔레트 생성을 사용하는지, 또는 초기 팔레트의 송신과 온 더 플라이 도출에 의한 초기 팔레트의 업데이팅의 조합을 시그널링할 수 있다. 일부 예들에서, 초기 팔레트는 최대 팔레트 사이즈에서의 전체 팔레트, 또는 감소된 사이즈 팔레트일 수도 있는데, 전체 팔레트의 경우 초기 팔레트에 있는 값들이 변경될 수 있고, 감소된 사이즈 팔레트의 경우, 값들은 초기 팔레트에 추가되고, 옵션으로는 초기 팔레트에 있는 값들이 변경된다.
- [0099] 위에서, 화소 값을 식별함으로써 맵을 송신하는 방법이 설명되었다. 위에서 설명된 그 방법과 함께, 맵의 송신은 라인 복사 (line copying) 를 시그널링함으로써 행해질 수 있다. 하나의 예에서, 라인 복사는 엔트리에 대한 화소 값이 상측 라인의 (또는 스캔이 수직이면 좌측 열에서의) 엔트리의 화소 값과 동일하도록 비디오 인코더 (20) 에 의해 시그널링된다. 그 다음에, 라인으로부터 복사되는 엔트리들의 '런'은 시그널링될 수도 있다. 또한, 엔트리가 복사되는 라인은 표시될 수 있으며; 상측의 여러 라인들이 이 목적을 위해 버퍼링될 수도 있다. 예를 들면, 이전의 네 개의 행들은 저장되고 어떤 행이 복사되는지는 절단된 단항 코드 또는 다른 코드들로 시그널링될 수도 있고, 그 다음에, 그 행의 얼마나 많은 엔트리들이 복사되는지가, 즉, 그 런은 시그널링될 수도 있다. 그런고로, 일부 예들에서, 엔트리에 대한 화소 값은 현재 행 바로 상측의 행 또는 둘 이상의 행들에 있는 엔트리의 화소 값과 동일하게 된다는 것이 시그널링될 수도 있다.
- [0100] 런이 시그널링되지 않는 경우, 그 런의 값은 상수/고정일 수도 있거나 또는 위에서 설명된 방법을 사용하여 사이드 정보에 의존적일 (그리고 디코더에 의해 도출될) 수도 있다.
- [0101] 그 맵은 송신될 필요가 없다는 것이 또한 가능하다. 예를 들어, 런은 특정한 포지션들에서만 시작할 수도 있다. 시작 포지션은 고정일 수도 있거나 또는 사이드 정보에 의존적일 (그리고 디코더에 의해 도출될) 수도 있어서, 시작 포지션의 시그널링은 스kip될 수도 있다. 대신, 위에서 설명된 기법들 중 하나 이상이 적용될 수도 있다. 암시적 시작 포지션 도출과 암시적 런 도출은 위에서 설명된 바와 동일한 방법을 사용하여 또한 결합될 수도 있다.
- [0102] 맵 송신의 둘 다의 방법들이 사용된다면, 화소가 팔레트로부터 획득되는지 또는 이전의 라인들로부터 획득되는지를 플래그가 나타낼 수도 있고, 그 다음에 인덱스는 팔레트 또는 행에 있는 엔트리, 그리고 마지막으로 '런'을 나타낸다.
- [0103] 도 2는 본 개시물의 팔레트 코딩 기법들을 구현할 수도 있는 일 예의 비디오 인코더 (20) 를 도시하는 블록도이다. 도 2는 설명의 목적으로 제공되고 본 개시물에서 폭넓게 예시되고 설명된 바와 같은 기법들의 제한으로서 고려되지 않아야 한다. 설명의 목적으로, 본 개시물은 HEVC 코딩의 맥락에서 비디오 인코더 (20) 를 설명한다. 그러나, 본 개시물의 기법들은 다른 코딩 표준들 또는 방법들에 적용 가능할 수도 있다.
- [0104] 비디오 인코더 (20) 는 본 개시물에서 설명되는 다양한 예들에 따라 팔레트 기반 비디오 코딩을 위한 기법들을 수행하도록 구성될 수도 있는 디바이스의 일 예를 나타낸다. 예를 들어, 비디오 인코더 (20) 는 팔레트 기반 코딩 또는 비-팔레트 기반 코딩 중 어느 하나를 사용하여 비디오 데이터의 다양한 블록들, 이를테면 HEVC 코딩에서의 CU들 또는 PU들을, 선택적으로 코딩하도록 구성될 수도 있다.
- [0105] 도 2의 예에서, 비디오 인코더 (20) 는 예측 프로세싱 부 (100), 잔차 생성 부 (102), 변환 프로세싱 부 (104), 양자화 부 (106), 역 양자화 부 (108), 역 변환 프로세싱 부 (110), 복원 부 (112), 필터 부 (114), 디코딩된 픽처 버퍼 (decoded picture buffer, DPB) (116), 비디오 메모리 (119), 및 엔트로피 인코딩 부 (118) 를 포함한다. 예측 프로세싱 부 (100) 는 인터 예측 프로세싱 부 (120) 와 인트라 예측 프로세싱 부 (126) 를 구비한다. 인터 예측 프로세싱 부 (120) 는 모션 추정 부와 모션 보상 부 (미도시) 를 구비한다. 비디오 인코더 (20) 는 본 개시물에서 설명되는 팔레트 기반 코딩 기법들의 다양한 양태들을 수행하도록 구성되는 팔레트 기반 인코딩 부 (122) 를 또한 구비한다. 다른 예들에서, 비디오 인코더 (20) 는 더 많거나, 더 적거나, 또는 상이한 기능성 컴포넌트들을 포함할 수도 있다.
- [0106] 도 4 내지 도 9를 참조하여 아래에서 더 상세히 설명될 바와 같이, 비디오 인코더 (20) 의 팔레트 기반 인코딩

부 (122)는 비디오 데이터의 현재 블록에 대한 팔레트를 생성하고, 비디오 데이터의 현재 블록에 대한 팔레트에 대한 이진 예측 벡터를 생성하는 것으로서, 이진 예측 벡터는 이전에 사용된 팔레트 엔트리들이 비디오 데이터의 현재 블록에 대한 팔레트에 재사용되는지의 여부를 나타내는 엔트리들을 포함하는, 상기 비디오 데이터의 현재 블록에 대한 팔레트에 대한 이진 예측 벡터를 생성하고, 런-길이 인코딩 기법을 사용하여 이진 예측 벡터를 인코딩하고, 그 팔레트를 사용하여 비디오 데이터의 현재 블록을 인코딩하도록 구성될 수도 있다.

[0107] 비디오 인코더 (20)는 비디오 데이터를 수신할 수도 있다. 비디오 인코더 (20)는 비디오 데이터의 픽처의 슬라이스에서의 각각의 CTU를 인코딩할 수도 있다. CTU들의 각각은 픽처의 동일 사이즈로 된 루마 코딩 트리 블록들 (CTB들) 및 대응하는 CTB들에 연관될 수도 있다. CTU를 인코딩하는 부분으로서, 예측 프로세싱 부 (100)은 큐드트리 구획화를 수행하여 CTU의 CTB들을 점차적으로 더 작은 블록들이 되게 분할할 수도 있다. 더 작은 블록은 CU들의 코딩 블록들일 수도 있다. 예를 들어, 예측 프로세싱 부 (100)는 CTU에 연관된 CTB를 네 개의 동일 사이즈로 된 서브-블록들로 구획화하며, 그 서브-블록들 중 하나 이상을 네 개의 동일 사이즈로 된 서브 서브-블록들로 구획화하는 등등을 수행할 수도 있다.

[0108] 도 2에 도시된 바와 같이, 비디오 메모리 (119)는 비디오 프레임 내의 현재 비디오 블록을 인코딩하기 위해 사용되는 비디오 데이터를 수신한다. 비디오 메모리 (119)는 (예컨대, 비디오 데이터를 저장하도록 구성되는) 비디오 인코더 (20)의 컴포넌트들에 의해 인코딩될 비디오 데이터를 저장할 수도 있다. 비디오 메모리 (119)에 저장된 비디오 데이터는, 예를 들어, 도 1의 비디오 소스 (18)로부터 획득될 수도 있다. DPB (116)는 비디오 인코더 (20)에 의해 비디오 데이터를 인코딩함에 있어서 (예컨대, 인트라 예측 또는 인터 예측 코딩 모드들이라고 또한 지칭되는 인트라 코딩 또는 인터 코딩 모드들에서) 사용하기 위한 참조 비디오 데이터를 저장하는 DPB의 하나의 예이다. 비디오 메모리 (119)와 DPB (116)는 동기식 DRAM (SDRAM)을 포함한 다이나믹 랜덤 액세스 메모리 (dynamic random access memory, DRAM), 자기저항 RAM (MRAM), 저항 RAM (RRAM), 또는 다른 유형들의 메모리 디바이스들과 같은 다양한 메모리 디바이스들 중 임의의 것에 의해 형성될 수도 있다. 비디오 메모리 (119)와 DPB (116)는 동일한 메모리 디바이스 또는 별개의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, 비디오 메모리 (119)는 비디오 인코더 (20)의 다른 컴포넌트들과 온-칩, 또는 그들 컴포넌트들에 대하여 오프-칩일 수도 있다.

[0109] 비디오 인코더 (20)는 CTU의 CU들을 인코딩하여 CU들의 인코딩된 표현들 (즉, 코딩된 CU들)을 생성할 수도 있다. CU를 인코딩하는 부분으로서, 예측 프로세싱 부 (100)는 CU의 하나 이상의 PU들 중에서 CU에 연관된 코딩 블록들을 구획화할 수도 있다. 따라서, 각각의 PU는 루마 예측 블록 및 대응하는 크로마 예측 블록들에 연관될 수도 있다. 비디오 인코더 (20)와 비디오 디코더 (30)는 다양한 사이즈들을 갖는 PU들을 지원할 수도 있다. 위에서 나타낸 바와 같이, CU의 사이즈는 CU의 루마 코딩 블록의 사이즈를 지칭할 수도 있고 PU의 사이즈는 PU의 루마 예측 블록의 사이즈를 지칭할 수도 있다. 특정 CU의 사이즈가 $2Nx2N$ 이라고 가정하면, 비디오 인코더 (20)와 비디오 디코더 (30)는 인트라 예측을 위한 $2Nx2N$ 또는 NxN 의 PU 사이즈들과, 인터 예측을 위한 $2Nx2N$, $2NxN$, $Nx2N$, NxN , 또는 유사한 것의 대칭적 PU 사이즈들을 지원할 수도 있다. 비디오 인코더 (20)와 비디오 디코더 (30)는 인터 예측을 위해 $2NxN$, $2NxN$, $nLx2N$, 및 $nRx2N$ 의 PU 사이즈들에 대한 비대칭 구획화를 또한 지원할 수도 있다.

[0110] 인터 예측 프로세싱 부 (120)는 CU의 각각의 PU에 대해 인터 예측을 수행함으로써 PU에 대한 예측 데이터를 생성할 수도 있다. PU에 대한 예측 데이터는 PU의 예측 블록들 및 그 PU에 대한 모션 정보를 포함할 수도 있다. 인터 예측 부 (121)는 PU가 I 슬라이스, P 슬라이스, 또는 B 슬라이스 중 어느 것에 있는지에 의존하여 CU의 PU에 대해 상이한 동작들을 수행할 수도 있다. I 슬라이스에서, 모든 PU들이 인트라 예측된다. 그런고로, PU가 I 슬라이스에 있으면, 인터 예측 부 (121)는 PU에 대해 인터 예측을 수행하지 않는다. 따라서, I-모드에서 인코딩된 블록들에 대해, 예측된 블록은 동일한 프레임 내의 이전에 인코딩된 이웃 블록들로부터 공간적 예측을 사용하여 형성된다.

[0111] PU가 P 슬라이스에 있다면, 인터 예측 프로세싱 부 (120)의 모션 추정 부는 PU에 대한 참조 지역을 위해 참조 픽처들의 리스트 (예컨대, "RefPicList0")에서의 참조 픽처들을 검색할 수도 있다. PU에 대한 참조 지역은, 참조 픽처 내의, PU의 샘플 블록들에 가장 밀접하게 대응하는 샘플 블록들을 포함하는 지역일 수도 있다. 모션 추정 부는 PU에 대한 참조 지역을 포함하는 참조 픽처의 RefPicList0에서의 포지션을 나타내는 참조 인덱스를 생성할 수도 있다. 덧붙여서, 모션 추정 부는 PU의 코딩 블록 및 참조 지역에 연관된 참조 케이션 사이의 공간적 변위를 나타내는 MV를 생성할 수도 있다. 예를 들면, MV는 현재 코딩된 픽처에서의 좌표들로부터 참조 픽처에서의 좌표들로의 오프셋을 제공하는 2차원 벡터일 수도 있다. 모션 추정 부는 참조 인덱스 및 MV를 PU의 모션 정보로서 출력할 수도 있다. 인터 예측 프로세싱 부 (120)의 모션 보상 부는

PU의 모션 벡터에 의해 나타내어진 참조 로케이션에서의 실제 또는 보간된 샘플들에 기초하여 PU의 예측 블록들을 생성할 수도 있다.

[0112] PU가 B 슬라이스에 있다면, 모션 추정 부는 PU에 대해 단-예측 또는 양-예측을 수행할 수도 있다. PU에 대한 단-예측을 수행하기 위해, 모션 추정 부는 PU에 대한 참조 지역을 위해 RefPicList0 또는 제 2 참조 픽처 리스트 ("RefPicList1")의 참조 픽처들을 검색할 수도 있다. 모션 추정 부는, PU의 모션 정보로서, 참조 지역을 포함하는 참조 픽처의 RefPicList0 또는 RefPicList1에서의 포지션을 나타내는 참조 인덱스, PU의 예측 블록 및 참조 지역에 연관된 참조 로케이션 사이의 공간적 변위를 나타내는 MV, 및 참조 픽처가 RefPicList0에 있는지 또는 RefPicList1에 있는지를 나타내는 하나 이상의 예측 방향 표시자들을 출력할 수도 있다. 인터 예측 프로세싱 부 (120)의 모션 보상 부는 PU의 모션 벡터에 의해 가리켜진 참조 지역에 있는 실제 또는 보간된 샘플들에 적어도 부분적으로 기초하여 PU의 예측 블록들을 생성할 수도 있다.

[0113] PU에 대한 양방향 인터 예측을 수행하기 위해, 모션 추정 부는 그 PU에 대한 참조 지역을 위해 RefPicList0에서의 참조 픽처들을 검색할 수도 있고, 또한 그 PU에 대한 다른 참조 지역을 위해 RefPicList1에서의 참조 픽처들을 검색할 수도 있다. 모션 추정 부는 참조 지역들을 포함하는 참조 픽처들의 RefPicList0 및 RefPicList1에서의 포지션들을 나타내는 참조 인덱스들을 생성할 수도 있다. 덧붙여서, 모션 추정 부는 참조 지역들에 연관된 참조 로케이션 및 PU의 샘플 블록 사이의 공간적 변위들을 나타내는 MV들을 생성할 수도 있다. PU의 모션 정보는 PU의 MV들 및 참조 인덱스들을 포함할 수도 있다. 모션 보상 부는 PU의 모션 벡터들에 의해 나타내어진 참조 지역들에 있는 실제 또는 보간된 샘플들에 적어도 부분적으로 기초하여 PU의 예측 블록들을 생성할 수도 있다.

[0114] 인트라 예측 프로세싱 부 (126)는 PU에 대해 인트라 예측을 수행함으로써 그 PU에 대한 예측 데이터를 생성할 수도 있다. PU에 대한 예측 데이터는 PU에 대한 예측 블록들과 다양한 선택스 엘리먼트들을 포함할 수도 있다. 인트라 예측 프로세싱 부 (126)는 I 슬라이스들, P 슬라이스들, 및 B 슬라이스들에서의 PU들에 대해 인트라 예측을 수행할 수도 있다.

[0115] PU에 대해 인트라 예측을 수행하기 위해, 인트라 예측 프로세싱 부 (126)는 PU에 대한 예측 데이터의 다수의 세트들을 생성하기 위해 다수의 인트라 예측 모드들을 사용할 수도 있다. 인트라 예측 프로세싱 부 (126)는 이웃하는 PU들의 샘플 블록들로부터의 샘플들을 사용하여 PU에 대한 예측 블록을 생성할 수도 있다. PU들, CU들, 및 CTU들에 대한 좌측에서 우측으로, 상단에서 하단으로의 인코딩 순서를 가정하면, 이웃하는 PU들은 PU의 상측, 우상측, 좌상측, 또는 좌측에 있을 수도 있다. 인트라 예측 프로세싱 부 (126)는 다양한 수들의 인트라 예측 모드들, 예컨대, 33 개의 방향성 인트라 예측 모드들을 사용할 수도 있다. 일부 예들에서, 인트라 예측 모드들의 수는 PU에 연관된 지역의 사이즈에 의존할 수도 있다.

[0116] 예측 프로세싱 부 (100)는 CU의 PU들에 대한 예측 데이터를, 그 PU들에 대해 인터 예측 프로세싱 부 (120)에 의해 생성된 예측 데이터 또는 그 PU들에 대해 인트라 예측 프로세싱 부 (126)에 의해 생성된 예측 데이터 중에서 선택할 수도 있다. 일부 예들에서, 예측 프로세싱 부 (100)는 예측 데이터의 세트들의 레이트/왜곡 메트릭들에 기초하여 CU의 PU들에 대한 예측 데이터를 선택한다. 선택된 예측 데이터의 예측 블록들은 본원에서는 선택된 예측 블록들이라고 지칭될 수도 있다.

[0117] 잔차 생성 부 (102)는, CU의 코딩 블록 (예컨대, 루마, Cb 및 Cr 코딩 블록)과 CU의 PU들의 선택된 예측 블록들 (예컨대, 예측 루마, Cb 및 Cr 블록들)에 기초하여, CU의 잔차 블록 (예컨대, 루마, Cb 및 Cr 잔차 블록)을 생성할 수도 있다. 예를 들면, 잔차 생성 부 (102)는 CU의 잔차 블록들에서의 각각의 샘플이 그 CU의 코딩 블록에서의 샘플 및 그 CU의 PU의 대응하는 선택된 예측 블록에서의 대응하는 샘플 사이의 차이와 동일한 값을 가지도록 CU의 잔차 블록들을 생성할 수도 있다.

[0118] 변환 프로세싱 부 (104)는 쿼드트리 구획화를 수행하여 CU에 연관된 잔차 블록들을 그 CU의 TU들에 연관된 변환 블록들로 구획화할 수도 있다. 따라서, TU가 루마 변환 블록 및 두 개의 크로마 변환 블록들에 대응할 수도 있다. CU의 TU들의 루마 및 크로마 변환 블록들의 사이즈들 및 포지션들은 그 CU의 PU들의 예측 블록들의 사이즈들 및 포지션들에 기초할 수도 있거나 또는 기초하지 않을 수도 있다. "잔차 쿼드 트리 (residual quad-tree)" (RQT)로서 알려진 쿼드트리 구조가 지역들의 각각에 대응하는 노드들을 포함할 수도 있다. CU의 TU들은 RQT의 리프 노드들에 대응할 수도 있다.

[0119] 변환 프로세싱 부 (104)는 CU의 각각의 TU에 대한 변환 계수 블록들을, 하나 이상의 변환들을 그 TU의 변환 블록들에 적용함으로써 생성할 수도 있다. 변환 프로세싱 부 (104)는 다양한 변환들을 TU에 연관된 변환 블

록에 적용할 수도 있다. 예를 들어, 변환 프로세싱 부 (104) 는 이산 코사인 변환 (discrete cosine transform, DCT), 방향성 변환, 또는 개념적으로 유사한 변환을 변환 블록에 적용할 수도 있다. 일부 예들에서, 변환 프로세싱 부 (104) 는 변환들을 변환 블록에 적용하지 않는다. 그런 예들에서, 변환 블록은 변환 계수 블록으로서 다루어질 수도 있다.

[0120] 양자화 부 (106) 는 계수 블록에서의 변환 계수들을 양자화할 수도 있다. 양자화 프로세스는 변환 계수들의 일부 또는 전부에 연관된 비트 깊이를 감소시킬 수도 있다. 예를 들어, n -비트 변환 계수가 양자화 동안에 m -비트 변환 계수로 버림될 (rounded down) 수도 있으며, 여기서 n 은 m 보다 크다. 양자화 부 (106) 는 CU에 연관된 양자화 파라미터 (QP) 값에 기초하여 그 CU의 TU에 연관된 계수 블록을 양자화할 수도 있다. 비디오 인코더 (20) 는 CU에 연관된 QP 값을 조정함으로써 그 CU에 연관된 계수 블록들에 적용되는 양자화 정도를 조정할 수도 있다. 양자화는 정보의 손실을 초래할 수도 있고, 따라서 양자화된 변환 계수들은 원래의 것들 보다 더 낮은 정밀도를 가질 수도 있다.

[0121] 역 양자화 부 (108) 와 역 변환 프로세싱 부 (110) 는 계수 블록으로부터 잔차 블록을 복원하기 위해 역 양자화 및 역 변환들을 계수 블록에 각각 적용할 수도 있다. 복원 부 (112) 는 TU에 연관된 복원된 변환 블록을 생성하기 위해 복원된 잔차 블록을 예측 프로세싱 부 (100) 에 의해 생성된 하나 이상의 예측 블록들로부터의 대응하는 샘플들에 가산할 수도 있다. CU의 각각의 TU에 대한 변환 블록들을 이런 식으로 복원함으로써, 비디오 인코더 (20) 는 CU의 코딩 블록들을 복원할 수도 있다.

[0122] 필터 부 (114) 는 하나 이상의 블록화제거 (deblocking) 동작들을 수행하여 CU에 연관된 코딩 블록들에서의 블록화 아티팩트들을 감소시킬 수도 있다. 디코딩된 픽처 버퍼 (116) 는, 필터 부 (114) 가 복원된 코딩 블록들에 대해 하나 이상의 블록화제거 동작들을 수행한 후에 복원된 코딩 블록들을 저장할 수도 있다. 인터 예측 프로세싱 부 (120) 는 다른 픽처들의 PU들에 대해 인터 예측을 수행하기 위해 복원된 코딩 블록들을 포함하는 참조 픽처를 사용할 수도 있다. 덧붙여서, 인트라 예측 프로세싱 부 (126) 는 CU와 동일한 픽처에서의 다른 PU들에 대해 인트라 예측을 수행하기 위해 디코딩된 픽처 버퍼 (116) 에서의 복원된 코딩 블록들을 사용할 수도 있다.

[0123] 엔트로피 인코딩 부 (118) 는 비디오 인코더 (20) 의 다른 기능성 컴포넌트들로부터 데이터를 수신할 수도 있다. 예를 들어, 엔트로피 인코딩 부 (118) 는 양자화 부 (106) 로부터 계수 블록들을 수신할 수도 있고 예측 프로세싱 부 (100) 로부터 신택스 엘리먼트들을 수신할 수도 있다. 엔트로피 인코딩 부 (118) 는 그 데이터에 대해 하나 이상의 엔트로피 인코딩 동작들을 수행하여 엔트로피 인코딩된 데이터를 생성할 수도 있다. 예를 들어, 엔트로피 인코딩 부 (118) 는 그 데이터에 대해 콘텍스트 적응 가변 길이 코딩 (context-adaptive variable length coding, CAVLC) 동작, CABAC 동작, 가변 대 가변 (variable-to-variable, V2V) 길이 코딩 동작, 신택스 기반 콘텍스트 적응 이진 산술 코딩 (syntax-based context-adaptive binary arithmetic coding, SBAC) 동작, 확률 간격 구획화 엔트로피 (Probability Interval Partitioning Entropy, PIPE) 코딩 동작, 지수-골롬 인코딩 동작, 또는 다른 유형의 엔트로피 인코딩 동작을 수행할 수도 있다. 비디오 인코더 (20) 는 엔트로피 인코딩 부 (118) 에 의해 생성된 엔트로피 인코딩된 데이터를 포함하는 비트스트림을 출력할 수도 있다. 예를 들면, 그 비트스트림은 CU에 대한 RQT를 표현하는 데이터를 포함할 수도 있다.

[0124] 도 3은 본 개시물의 기법들을 구현하도록 구성되는 일 예의 비디오 디코더 (30) 를 도시하는 블록도이다. 도 3은 설명의 목적으로 제공되고 본 개시물에서 폭넓게 예시되고 설명된 바와 같은 기법들로 제한하고 있지는 않다. 설명의 목적으로, 본 개시물은 HEVC 코딩의 맥락에서 비디오 디코더 (30) 를 설명한다. 그러나, 본 개시물의 기법들은 다른 코딩 표준들 또는 방법들에 적용 가능할 수도 있다.

[0125] 도 3의 예에서, 비디오 디코더 (30) 는 엔트로피 디코딩 부 (150), 예측 프로세싱 부 (152), 역 양자화 부 (154), 역 변환 프로세싱 부 (156), 복원 부 (158), 필터 부 (160), 비디오 메모리 (163), 및 디코딩된 픽처 버퍼 (DPB) (162) 를 구비한다. 예측 프로세싱 부 (152) 는 모션 보상 부 (164) 와 인트라 예측 프로세싱 부 (166) 를 구비한다. 비디오 디코더 (30) 는 또한, 본 개시물에서 설명되는 팔레트 기반 코딩 기법들의 다양한 양태들을 수행하도록 구성되는 팔레트 기반 디코딩 부 (165) 를 구비한다. 다른 예들에서, 비디오 디코더 (30) 는 더 많거나, 더 적거나, 또는 상이한 기능성 컴포넌트들을 포함할 수도 있다.

[0126] 도 4 내지 도 9를 참조하여 아래에서 더 상세히 설명될 바와 같이, 비디오 디코더 (30) 의 팔레트 기반 디코딩 부 (165) 는 비디오 데이터의 현재 블록에 대한 인코딩된 이진 예측 벡터 (예컨대, 런-길이 인코딩으로 인코딩 된 이진 예측 벡터) 를 수신하고, 런-길이 디코딩 기법을 사용하여 인코딩된 이진 예측 벡터를 디코딩하고, 이진 예측 벡터에 기초하여 비디오 데이터의 현재 블록에 대한 팔레트를 생성하는 것으로서, 이진 예측 벡터는 이

전에 사용된 팔레트 엔트리들이 비디오 데이터의 현재 블록에 대한 팔레트에 재사용되는지의 여부를 나타내는 엔트리들을 포함하는, 상기 비디오 데이터의 현재 블록에 대한 팔레트를 생성하고, 그리고 그 팔레트를 사용하여 비디오 데이터의 현재 블록을 디코딩하도록 구성될 수도 있다.

[0127] 코딩된 픽처 베퍼 (CPB), 예컨대, 비디오 메모리 (163) 가 비트스트림의 인코딩된 비디오 데이터 (예컨대, NAL 유닛들) 를 수신 및 저장할 수도 있다. 엔트로피 디코딩 부 (150) 는 CPB로부터 인코딩된 비디오 데이터 (예컨대, NAL 유닛들) 를 수신하고 신택스 엘리먼트들을 디코딩하기 위해 그 NAL 유닛들을 파싱할 수도 있다.

엔트로피 디코딩 부 (150) 는 NAL 유닛들에서의 엔트로피 인코딩된 신택스 엘리먼트들을 엔트로피 디코딩할 수도 있다. 예측 프로세싱 부 (152), 역 양자화 부 (154), 역 변환 프로세싱 부 (156), 복원 부 (158), 및 필터 부 (160) 는 비트스트림으로부터 추출된 신택스 엘리먼트들에 기초하여 디코딩된 비디오 데이터를 생성할 수도 있다.

[0128] 도 3에 도시된 바와 같이, 비디오 메모리 (163) 는 비디오 프레임 내의 현재 비디오 블록을 디코딩하기 위해 사용되는 비디오 데이터를 수신한다. 비디오 메모리 (163) 는 (예컨대, 비디오 데이터를 저장하도록 구성된) 비디오 디코더 (30) 의 컴포넌트들에 의해 디코딩될 비디오 데이터를 저장할 수도 있다. 비디오 메모리 (163) 에 저장된 비디오 데이터는, 예를 들어, 비디오 인코더 (20) 에 의해 생성된 인코딩된 비디오 비트스트림으로부터 획득될 수도 있다. DPB (162) 는 비디오 디코더 (30) 에 의해 비디오 데이터를 디코딩함에 있어서 (예컨대, 인트라 예측 또는 인터 예측 코딩 모드들이라고 또한 지칭되는 인트라 코딩 또는 인터 코딩 모드들에서) 사용하기 위한 참조 비디오 데이터를 저장하는 DPB의 하나의 예이다. 비디오 메모리 (163) 와 DPB (162) 는 동기식 DRAM (SDRAM) 을 포함한 다이나믹 랜덤 액세스 메모리 (DRAM), 자기저항 RAM (MRAM), 저항 RAM (RRAM), 또는 다른 유형들의 메모리 디바이스들과 같은 다양한 메모리 디바이스들 중 임의의 것에 의해 형성될 수도 있다. 비디오 메모리 (163) 와 DPB (162) 는 동일한 메모리 디바이스 또는 별개의 메모리 디바이스들에 의해 제공될 수도 있다. 다양한 예들에서, 비디오 메모리 (163) 는 비디오 디코더 (30) 의 다른 컴포넌트들과 온-칩, 또는 그들 컴포넌트들에 대하여 오프-칩일 수도 있다.

[0129] 비트스트림의 NAL 유닛들은 코딩된 슬라이스 NAL 유닛들을 포함할 수도 있다. 비트스트림을 디코딩하는 부분으로서, 엔트로피 디코딩 부 (150) 는 코딩된 슬라이스 NAL 유닛들로부터 신택스 엘리먼트들을 추출하고 엔트로피 디코딩할 수도 있다. 코딩된 슬라이스들의 각각은 슬라이스 헤더 및 슬라이스 데이터를 포함할 수도 있다. 슬라이스 헤더는 슬라이스에 관계된 신택스 엘리먼트들을 포함할 수도 있다. 슬라이스 헤더에서의 신택스 엘리먼트들은 슬라이스를 포함하는 픽처에 연관된 PPS를 식별하는 신택스 엘리먼트를 포함할 수도 있다.

[0130] 비트스트림으로부터 신택스 엘리먼트들을 디코딩하는 것 외에도, 비디오 디코더 (30) 는 비구획화된 CU에 대해 복원 동작을 수행할 수도 있다. 비구획화된 CU에 대해 복원 동작을 수행하기 위해, 비디오 디코더 (30) 는 CU의 각각의 TU에 대해 복원 동작을 수행할 수도 있다. CU의 각각의 TU에 대한 복원 동작을 수행함으로써, 비디오 디코더 (30) 는 CU의 잔차 블록들을 복원할 수도 있다.

[0131] CU의 TU에 대해 복원 동작을 수행하는 부분으로서, 역 양자화 부 (154) 는 TU에 연관된 계수 블록들을 역 양자화, 즉, 탈양자화 (de-quantization) 할 수도 있다. 역 양자화 부 (154) 는 TU의 CU에 연관된 QP 값을 사용하여 양자화 정도와, 비슷하게, 역 양자화 부 (154) 에 대해 적용할 역 양자화 정도를 결정할 수도 있다. 다시 말하면, 압축 비율, 즉, 원래의 시퀀스 및 압축된 시퀀스를 표현하는데 사용된 비트들의 수의 비율은, 변환 계수들을 양자화하는 경우에 사용되는 QP의 값을 조정함으로써 제어될 수도 있다. 압축 비율은 채용된 엔트로피 코딩하는 방법에 또한 의존할 수도 있다.

[0132] 역 양자화 부 (154) 가 계수 블록을 역 양자화한 후, 역 변환 프로세싱 부 (156) 는 TU에 연관된 잔차 블록을 생성하기 위하여 하나 이상의 역 변환들을 계수 블록에 적용할 수도 있다. 예를 들어, 역 변환 프로세싱 부 (156) 는 역 DCT, 역 정수 변환, 역 카루넨-뢰베 변환 (Karhunen-Loeve transform, KLT), 역 회전 변환, 역 방향성 변환, 또는 다른 역 변환을 계수 블록에 적용할 수도 있다.

[0133] PU가 인트라 예측을 사용하여 인코딩되면, 인트라 예측 프로세싱 부 (166) 는 PU에 대한 예측 블록들을 생성하기 위해 인트라 예측을 수행할 수도 있다. 인트라 예측 프로세싱 부 (166) 는 인트라 예측 모드를 사용하여 공간적으로 이웃하는 PU들의 예측 블록들에 기초하여 PU에 대한 예측 루마, Cb 및 Cr 블록들을 생성할 수도 있다. 인트라 예측 프로세싱 부 (166) 는 비트스트림으로부터 디코딩된 하나 이상의 신택스 엘리먼트들에 기초하여 PU에 대한 인트라 예측 모드를 결정할 수도 있다.

- [0134] 예측 프로세싱 부 (152) 는 비트스트림으로부터 추출된 십екс 앤리먼트들에 기초하여 제 1 참조 픽처 리스트 (RefPicList0) 및 제 2 참조 픽처 리스트 (RefPicList1) 를 구축할 수도 있다. 더욱이, PU가 인터 예측을 사용하여 인코딩되면, 엔트로피 디코딩 부 (150) 는 그 PU에 대한 모션 정보를 추출할 수도 있다. 모션 보상 부 (164) 는, PU의 모션 정보에 기초하여, PU에 대한 하나 이상의 참조 지역들을 결정할 수도 있다. 모션 보상 부 (164) 는, PU에 대한 하나 이상의 참조 블록들에서의 샘플 블록들에 기초하여, PU에 대한 예측 블록들 (예컨대, 예측 루마, Cb 및 Cr 블록들) 을 생성할 수도 있다.
- [0135] 복원 부 (158) 는, 적용가능할 경우, CU의 TU들에 연관된 변환 블록들 (예컨대, 루마, Cb 및 Cr 변환 블록들) 과 그 CU의 PU들의 예측 블록들 (예컨대, 예측 루마, Cb 및 Cr 블록들), 즉 인트라 예측 데이터 또는 인터 예측 데이터 중 어느 하나를 사용하여 그 CU의 코딩 블록들 (예컨대, 루마, Cb 및 Cr 코딩 블록들) 을 복원할 수도 있다. 예를 들어, 복원 부 (158) 는 변환 블록들 (예컨대, 루마, Cb 및 Cr 변환 블록들) 의 샘플들을 예측 블록들 (예컨대, 예측 루마, Cb 및 Cr 블록들) 의 대응하는 샘플들에 가산하여 CU의 코딩 블록들 (예컨대, 루마, Cb 및 Cr 코딩 블록들) 을 복원할 수도 있다.
- [0136] 필터 부 (160) 는 블록화제거 동작을 수행하여 CU의 코딩 블록들 (예컨대, 루마, Cb 및 Cr 코딩 블록들) 에 연관된 블록화 아티팩트들을 감소시킬 수도 있다. 비디오 디코더 (30) 는 CU의 코딩 블록들 (예컨대, 루마, Cb 및 Cr 코딩 블록들) 을 디코딩된 픽처 베퍼 (162) 에 저장할 수도 있다. 디코딩된 픽처 베퍼 (162) 는 후속하는 모션 보상, 인트라 예측, 및 디스플레이 디바이스, 이를테면 도 1의 디스플레이 디바이스 (32) 상의 프레젠테이션을 위해 참조 픽처들을 제공할 수도 있다. 예를 들면, 비디오 디코더 (30) 는, 디코딩된 픽처 베퍼 (162) 에서의 블록들 (예컨대, 루마, Cb 및 Cr 블록들) 에 기초하여, 다른 CU들의 PU들에 대해 인트라 예측 또는 인터 예측 동작들을 수행할 수도 있다.
- [0137] 도 4는 비디오 인코더 (20) 의 팔레트 기반 인코딩 부 (122) 를 더 상세히 도시하는 블록도이다. 팔레트 기반 인코딩 부 (122) 는 이진 예측 백터를 인코딩하기 위해 본 개시물의 예의 기법들 중 하나 이상을 수행하도록 구성될 수도 있다.
- [0138] 위에서 설명된 바와 같이, 팔레트 기반 인코딩 부 (122) 는 비디오 데이터의 블록 (예컨대, CU 또는 PU) 을 팔레트 기반 인코딩 모드로 인코딩하도록 구성될 수도 있다. 팔레트 기반 인코딩 모드에서, 팔레트가 인덱스에 의해 번호 부여된 그리고 화소 값들을 나타내는데 사용될 수도 있는 컬러 성분 값들 (예를 들어, RGB, YUV 등) 또는 강도들을 표현하는 엔트리들을 포함할 수도 있다. 팔레트 생성 부 (203) 는 비디오 데이터의 현재 블록에 대한 화소 값들 (212) 을 수신하고 비디오 데이터의 현재 블록에 대한 컬러 값들의 팔레트를 생성하도록 구성될 수도 있다. 팔레트 생성 부 (203) 는 위에서 논의된 히스토그램-기반 기법들을 포함하는 비디오 데이터의 현재 블록에 대한 팔레트를 생성하기 위한 임의의 기법들을 사용할 수도 있다. 팔레트 생성 부 (203) 는 임의의 사이즈의 팔레트를 생성하도록 구성될 수도 있다. 하나의 예에서, 팔레트 생성 부 (203) 는 32 개의 팔레트 엔트리들을 생성하도록 구성될 수도 있는데, 각각의 팔레트 엔트리는 화소의 Y, Cr, 및 Cb 성분들에 대한 화소 값들을 포함한다. 전자의 예에서, 각각의 팔레트 엔트리가 샘플 (화소) 의 모든 컬러 성분들에 대한 값을 명시한다고 가정된다. 그러나, 이 문서에서 설명되는 개념들은 각각의 컬러 성분에 대해 별개의 팔레트를 사용하는 것에 적용 가능하다.
- [0139] 일단 팔레트가 팔레트 생성 부 (203) 에 의해 생성되면, 맵 부 (204) 는 비디오 데이터의 현재 블록에서의 특정 화소가 팔레트 생성 부 (203) 에 의해 생성된 팔레트에서의 엔트리에 의해 표현될 수도 있는지의 여부를 나타내는, 비디오 데이터의 현재 블록에 대한 맵을 생성할 수도 있다. 맵 부 (204) 는 어떻게 각각의 화소가 팔레트로부터의 엔트리들을 사용하는 (또는 사용하지 않는)지를 나타내는 십екс 앤리먼트들을 포함하는 맵 (214) 을 생성할 수도 있다. 비디오 데이터의 현재 블록에서의 화소에 대한 값이 팔레트에서 발견되지 않고 따라서 팔레트로의 인덱스에 의해 표현될 수 없다면, 맵 부 (204) 는 그 특정 화소에 대한 화소 값을 명시적으로 송신할 수도 있다. 일부 예들에서, 맵 부 (204) 는 팔레트에서 발견된 엔트리들 중 하나의 엔트리로부터 명시적 화소 값을 예측할 수도 있다. 일부 다른 예들에서, 맵 부 (204) 는 화소를 양자화하고 양자화된 값을 송신할 수도 있다.
- [0140] 블록에서의 화소들의 각각에 대해 사용되는 컬러 값들을 나타내는 십екс 앤리먼트들을 시그널링하는 것 외에도, 팔레트 기반 인코딩 부 (122) 는 비디오 데이터의 현재 블록을 위해 사용될 팔레트를 시그널링하도록 또한 구성될 수도 있다. 본 개시물의 기법들에 따라, 팔레트 기반 인코딩 부 (122) 는 비디오 데이터의 특정 블록에 대한 팔레트의 값을 나타내기 위해 시그널링되는 데이터의 양을 감소시키기 위해 팔레트 예측 기법들을 채용하도록 구성될 수도 있다.

- [0141] 팔레트 예측의 하나의 예로서, 2014년 6월 20일 현재 http://phenix.int-evry.fr/jct/doc_end_user/documents/17_Valencia/wg11/JCTVC-Q0094-v1.zip으로부터 입수 가능한 JCTVC-Q0094에서 설명된 바와 같이, 팔레트가 예측자 팔레트로부터 복사되는 엔트리들을 포함할 수도 있다. 예측자 팔레트가 팔레트 모드를 사용하는 이전에 코딩된 블록들로부터의 또는 다른 복원된 샘플들로부터의 팔레트 엔트리들을 포함할 수도 있다. 도 4에 도시된 바와 같이, 팔레트 기반 인코딩 부 (122)는 예측자 팔레트 베퍼 (210)를 포함할 수도 있다. 예측자 팔레트 베퍼 (210)는 이전에 인코딩된 블록들로부터의 다수의 이전에 사용된 팔레트 엔트리들을 저장하도록 구성될 수도 있다. 하나의 예로서, 예측자 팔레트 베퍼 (210)는 미리 결정된 사이즈의 선입 선출 (first-in, first-out, FIFO) 베퍼로서 구성될 수도 있다. 예측자 팔레트 베퍼 (210)는 임의의 사이즈일 수도 있다. 하나의 예에서, 예측자 팔레트 베퍼 (210)는 64 개까지의 이전에 사용된 팔레트 엔트리들을 포함한다.
- [0142] 일부 예들에서, 팔레트 기반 인코딩 부 (122)는 예측자 팔레트 베퍼 (210)에서의 모든 팔레트 엔트리들이 고유하도록 예측자 팔레트 베퍼 (210)에서의 엔트리들을 프루닝 (pruning) 하도록 구성될 수도 있다. 다시 말하면, 예측자 팔레트 베퍼 (210)에 추가될 각각의 새로운 팔레트 엔트리에 대해, 팔레트 기반 인코딩 부 (122)는 예측자 팔레트 베퍼 (210)에 이미 저장된 다른 동일한 엔트리들이 없다는 것을 먼저 체크하도록 구성될 수도 있다. 동일한 엔트리들이 없다면, 새로운 팔레트 엔트리는 예측자 팔레트 베퍼 (210)에 추가된다. 새로운 엔트리가 현존 엔트리의 중복물이면, 새로운 팔레트 엔트리는 예측자 팔레트 베퍼 (210)에 추가되고 중복된 엔트리들은 예측자 팔레트 베퍼 (210)로부터 제거된다.
- [0143] 팔레트 기반 인코딩 부 (122)는, 팔레트 생성 부 (203)에 의해 생성된 비디오 데이터의 현재 블록에 대한 팔레트에서의 각각의 엔트리에 대해, 예측자 팔레트 베퍼 (210)에서의 팔레트 엔트리가 비디오 데이터의 현재 블록에 대한 팔레트에서의 엔트리들 중 하나에 복사되는지 (또는 재사용되는지)의 여부를 나타내는 (예컨대, 플래그 = 1에 의해 나타내어진) 이진 플래그를 생성 및 시그널링하도록 구성되는 이진 예측 벡터 생성 부 (206)를 포함할 수도 있다. 다시 말하면, 이진 예측자 벡터에서의 1의 값을 갖는 플래그는 예측자 팔레트 베퍼 (210)에서의 대응하는 엔트리가 현재 블록에 대한 팔레트를 위해 재사용된다는 것을 나타내는 반면, 이진 예측 벡터에서의 0의 값을 갖는 플래그는 예측자 팔레트 베퍼 (210)에서의 대응하는 엔트리가 현재 블록에 대한 팔레트에 대해 재사용되지 않는다는 것을 나타낸다. 덧붙여, 팔레트 기반 인코딩 부 (122)는 예측자 팔레트 베퍼 (210)에서의 엔트리들로부터 복사될 수 없는 현재 팔레트에 대한 일부 값을 명시적으로 시그널링하도록 구성될 수도 있다. 새로운 엔트리들의 수 역시 시그널링될 수도 있다.
- [0144] 2014년 3월 25일자로 출원된 미국 가출원 제61/970,257호; 2014년 4월 17일자로 출원된 미국 가출원 제61/981,105호; 및 2014년 5월 23일자로 출원된 미국 가출원 제62/002,668호에서, 이진 트리 기반 시그널링 방법과 말단 (end) -포지션 기반 시그널링 방법들이 팔레트 이진 예측자 벡터의 코딩을 위해 제안되었다. 2014년 5월 23일자로 출원된 미국 가출원 제62/002,741호에서, 그룹 기반 시그널링 방법이 제안되었다. 본 개시물은 이진 예측 벡터를 생성, 인코딩, 및 디코딩하는 추가적인 기법들을 제안한다.
- [0145] 본원에서 설명되는 일부 예들은 코딩 효율을 개선하기 위해 팔레트 예측 벡터를 코딩하는 방법들에 관련된다. 예를 들어, 이진 예측 벡터 생성 부 (206)에 의해 생성된 이진 예측 벡터가 다음에 의해 표시된다고 가정한다:
- [0146] $b = [b_0, b_1, \dots, b_{N-1}]$, $N \geq 0$, $b_i \in \{0,1\}$, $0 \leq i < N$
- [0147] 위의 수학식에서, $b_i \in \{0,1\}$, $0 \leq i < N$ 은 예측 플래그 (또한 이진 플래그 또는 이진 예측 플래그로 지칭됨)를 표시한다. $N=0$, $b = \phi$ (즉, b 가 앤프티 (empty) 벡터)이면, 그것은 시그널링될 필요가 없다. 그러므로, 다음의 설명에서, $N > 0$ 인 것으로 가정될 수도 있다.
- [0148] 도 5는 예측자 팔레트 베퍼 (210) 및 현재 팔레트 (220)의 하나의 예를 도시한다. 도 5에서 알 수 있듯이, 현재 팔레트 (220)는 엔트리 인덱스들 (1, 2, 5, 및 9)에 연관된 예측자 팔레트 베퍼 (210)로부터의 화소 값을 재사용한다. 그래서, 도 4의 이진 예측 벡터 생성 부 (206)에 의해 생성된 이진 예측자 벡터는 $b = [110010001000]$ 이 될 것이다. 이 예에서 알 수 있듯이, 이진 예측 벡터 b 는 예측자 팔레트 베퍼 (210)에서의 첫 번째, 두 번째, 다섯 번째, 및 아홉 번째 인덱스들에 대응하는 1의 값을 갖는 플래그들을 포함한다. 다시 말하면, 예측자 팔레트 베퍼 (210)에서의 첫 번째, 두 번째, 다섯 번째, 및 아홉 번째 엔트리들은 현재 팔레트 (220)에 대해서만 재사용되는 엔트리들이다. 현재 팔레트 (220)에서의 엔트리 인덱스들 (5~8)에 대해, 팔레트 기반 인코딩 부 (122)는 팔레트 엔트리 값들을 (예컨대, 명시적 시그널링 또는 다른 예측 기법을

사용하여) 인코딩된 비디오 비트스트림에서 시그널링하도록 구성될 수도 있다.

[0149] 본 개시물의 하나 이상의 기법들에 따라, 비디오 인코더 (20)는 인코딩된 비디오 비트스트림에서 팔레트를 시그널링하는데 필요한 데이터의 양을 줄이기 위하여 이진 예측자 벡터 b 를 인코딩 또는 대체적으로 인코딩하도록 구성될 수도 있다. 도 4에 도시된 바와 같이, 이진 예측 벡터 압축 부 (209)는 인코딩된 이진 예측 벡터 (215)를 생성 및 시그널링하도록 구성될 수도 있다. 그러나, 본 개시물의 이진 예측 벡터 압축 기법들은 엔트로피 인코딩 부 (118)를 포함하는 비디오 인코더 (20)의 다른 구조들에서 구현될 수도 있다는 것이 이해되어야 한다.

[0150] 본 개시물의 하나의 예에서, 이진 예측 벡터 압축 부 (209)는 런-길이 기반 인코딩 기법들을 사용하여 이진 예측 벡터를 인코딩하도록 구성될 수도 있다. 예를 들어, 이진 예측 벡터 압축 부 (209)는 지수-골롬 코드를 사용하여 이진 예측 벡터에서의 '1들' 간의 연속적인 '0들'의 수를 시그널링함으로써 이진 예측 벡터를 인코딩하도록 구성될 수도 있다. 일 예로서, $b = [110010001000]$ 이라고 다시 가정한다. 이 예에서는, 도 6에 도시된 바와 같이, 이진 예측 벡터 (즉, b)는 다음으로서 표현될 수 있다: '영 개의 연속적인 0들'-'1'-'영 개의 연속적인 0들'-'1'-'두 (2) 개의 연속적인 0들'-'1'-'세 (3) 개의 연속적인 0들'-'1'- 그리고 '네 (4) 개의 연속적인 0들'. $b_i \in \{0,1\}$ 라는 것이 알려져 있기 때문에, 마지막 '연속적인 0' 그룹을 제외하면, 각각의 '연속적인 0' 그룹에는 '1'이 그 뒤를 따라야만 한다. 그러므로, 이진 예측 벡터 압축 부 (209)는 이진 예측 벡터 b 를 '영 개의 연속적인 0'-'영 개의 연속적인 0'-'두 개의 연속적인 0'-'세 개의 연속적인 0'-'네 개의 연속적인 0'으로서 표현하는 영 기반 런-길이 코딩 기법들을 사용할 수도 있는데, 그 이진 예측 벡터는 런-길이 시퀀스 '0-0-2-3-4'로서 표시될 수 있다.

[0151] 런-길이 기반 시그널링에 관련된 본 개시물의 하나 이상의 예들에 따라, 런-길이 시퀀스를 코딩하기 위해, 골롬-라이스 코드, 임의의 차수의 지수-골롬 코드, 절단된 지수-골롬 코드, 절단된-라이스 코드 또는 절단된 이진화들을 포함하는 임의의 다른 이진화들이 사용될 수도 있다. 하나의 예에서, 이진 예측 벡터 압축 부 (209)는 0차 지수-골롬 코드를 런-길이 코딩 기법으로서 사용한다.

[0152] 절단된 이진화에 대해, maxsymbol은 이진 벡터에서의 '1'의 포지션 및 이진 벡터 사이즈에 의존하는 런의 최대 가능 값일 수 있는데, 이진 벡터의 말단으로 이동할 시, 최대 가능 런 값은 그 벡터 내의 포지션에 의존하여 벡터 사이즈로부터 0으로 감소되기 때문이다. 예를 들어, 최대 심볼은 이진 벡터 길이 또는 이진 벡터 길이 마이너스 그 런이 카운트되고 있는 '1'의 포지션일 수 있다. 다르게 말하면, 그것은 이진 벡터의 말단에서부터 측정된 나머지 길이이다. 특정 사이즈, 예컨대, 13의 이진 벡터 b 를 갖는 위의 예에 대해, 런-길이 시퀀스 '0-0-2-3-4'는 절단된 이진화 '0[13]-0[12]-2[11]-3[8]-4[4]'로 코딩될 수 있으며, 여기서 최대 심볼은 격쇄팔호들 내에 제공된다.

[0153] 또한, 일부 예들에서, 이진화는 이진 벡터에서의 엘리먼트 (0 또는 1)의 포지션 또는 인덱스에 의존적일 수도 있다. 특정 예로서, 그 포지션이 특정 임계값보다 더 작다면, 하나의 유형의 이진화가 사용되며; 그렇지 않으면, 다른 유형의 이진화가 적용된다. 일부 예들에서, 이진화 유형은 상이한 이진화 코드들, 또는 동일한 코드 패밀리이지만 상이한 순서를 갖는, 이를테면 지수-골롬 코드이다.

[0154] 하나의 예에서, 그 임계값은 이전 블록 또는 이전 팔레트 코딩된 블록으로부터의 팔레트 길이일 수도 있다. 다른 예에서, 임계값은 어떤 디폴트 값으로 고정되거나 또는 블록, 슬라이스, 픽쳐 마다 또는 다른 곳에서 시그널링될 수 있다. 대응하는 기법이 런 값들을 코딩하기 위한 CABAC 콘텍스트를 정의하는데 옵션적으로 사용될 수도 있다는 것이 인식될 것이다. 덧붙여, 팔레트 기반 인코딩 부 (122)는 시그널링된 '1' 엘리먼트들의 수 (즉, 현재 팔레트 (220)에 대해 재사용되고 있는 것으로서 나타내어진 예측자 팔레트 버퍼 (210)로부터의 팔레트 엔트리들의 수) 가 최대 가능 수에 도달하는 경우 런-길이 시그널링을 정지시키도록 구성될 수도 있다. 일부 예들에서, 최대 가능 수는 최대 가능 팔레트 사이즈이다.

[0155] 본 개시물의 일부 예들은 이진 예측 벡터 b 를 나타내는 런-길이 시퀀스의 말단 포지션 코딩에 관련된다. 본 개시물의 하나 이상의 예들에서, 이진 예측 벡터 압축 부 (209)는 이진 예측 벡터의 말단 포지션을 코딩하기 위해 예약된 (reserved) 런-길이 L 를 사용하여 이진 예측 벡터 b 를 인코딩하도록 구성될 수도 있다. 하나의 예에서, $L=1$ 이 예약된 런-길이로서 사용된다. 비디오 인코더 (20)에서, 런-길이가 L 이상이면, 이진 예측 벡터 압축 부 (209)는 1을 런-길이에 더하도록 구성된다. 실제 런-길이가 L 미만이면, 이진 예측 벡터 압축 부 (209)는 런-길이를 그대로 시그널링하도록 구성된다. 이진 예측 벡터 압축 부 (209)는 예약된 런-길이 L 를 갖는 말단 포지션 런-길이를 시그널링할 수도 있다.

- [0156] 비슷하게, 비디오 디코더 (30)에서, 런-길이의 디코딩된 값이 L 보다 더 크면, 1이 실제 런-길이로부터 감산된다. 디코딩된 값 또는 런-길이가 L 보다 더 작다면, 디코딩된 값은 실제 런-길이로서 사용된다. 디코딩된 값이 L 과 동일하면, 이진 예측 벡터 b 에서의 남아있는 포지션들은 모두 0이다. 그런고로, 디코딩된 값이 L 과 동일하면, 더 이상의 런 시그널링은 필요하지 않다.
- [0157] 위에서와 동일한 예 (즉, $b = [110010001000]$)를 사용하면 그리고 $L=1$ 이라고 가정하면, 이진 예측 벡터 압축부 (209)는 도 6의 런-길이 시퀀스 '0-0-2-3-4'를 '0-0-3-4-1'로서 시그널링하도록 구성된다. 그 다음에, 위의 규칙들을 적용하면, 비디오 디코더 (30)는 런-길이 시퀀스를 '0-0-2-3-말단'으로서 복원하도록 구성될 수도 있다. 다시 말하면, 0의 첫 번째 런-길이 값은 0으로서 디코딩되고 0의 다음 런-길이 시퀀스는 0으로서 디코딩되는데, 0 런-길이 시퀀스들의 양쪽 모두가 $L=1$ 의 예약된 런-길이 값보다 더 작기 때문이다. 다음 런-길이 시퀀스는 3이고, 그래서, 비디오 디코더 (30)는 3의 값에서 1을 감산하여 2를 획득하도록 구성될 것인데, 3의 수신된 값이 $L=1$ 의 예약된 런-길이 값보다 더 크기 때문이다. 비슷하게, 비디오 디코더 (30)는 다음 런-길이 시퀀스에 대해 4의 수신된 값에서 1을 감산하여 3을 획득하도록 구성될 것인데, 4의 수신된 값은 $L=1$ 의 예약된 런-길이 값보다 더 크기 때문이다. 마지막으로, 마지막 수신된 런-길이 값 1은 $L=1$ 의 예약된 런-길이 값과 동일하다. 따라서, 비디오 디코더 (30)는 '1'의 추가의 값들이 이진 예측 벡터에 존재하지 않는다고 결정할 수도 있다.
- [0158] 본 개시물의 다른 예에서, 이진 예측 벡터 압축 부 (209)는 이진 예측 벡터에서의 '1들'의 총 수 (즉, 예측자 팔레트 베퍼 (210)로부터 재사용된 팔레트 엔트리들의 표시들의 수)가 최대 가능 팔레트 사이즈보다 작은 경우 말단 포지션 코딩만을 적용하도록 구성될 수도 있다. 이진 예측 벡터에서의 '1들'의 총 수가 최대 가능 팔레트 사이즈와 동일하다면, 이진 예측 벡터 압축 부 (209)는 마지막 런-길이의 시그널링을 바이패스하도록 구성될 수도 있다.
- [0159] 위의 예에서, 비디오 디코더 (30)가 최대 가능 팔레트 사이즈가 4라고 결정한다면, '0-0-2-3'의 런-길이 시퀀스 (또는 위에서 설명된 예약된 런-길이 규칙들에 따른 런-길이 시퀀스 0-0-3-4)는 이진 예측 벡터 b 를 복원하는데 충분할 수도 있다. 일부 예들에서, 최대 가능 팔레트 사이즈는 미리 결정될 수도 있다. 따라서, 최대 가능 팔레트 사이즈는 비디오 인코더 (20)와 비디오 디코더 (30)에 의해 미리 결정될 수도 있다. 다른 예들에서, 최대 가능 팔레트 사이즈는 비디오 인코더 (20)에 의해 인코딩된 비디오 비트스트림에서 시그널링될 수도 있다. 최대 가능 팔레트 사이즈를 시그널링하는 것은 임의의 적합한 기준의 데이터 통신 기법들 또는 본원에서 설명되는 임의의 적합한 데이터 통신 기법들을 사용하여 수행될 수도 있다.
- [0160] 본 개시물의 다른 예에서, 이진 예측 벡터 압축 부 (209)는 런-길이 시퀀스에서의 마지막 런-길이가 0이 아닌 경우에만 위에서 설명된 말단 포지션 코딩 기법들을 적용하도록 구성될 수도 있다. 런-길이 시퀀스에서의 마지막 런-길이가 0이면, 말단 포지션 인코딩이 바이패스될 수 있다. 이 명제 (statement)는 이진 예측 벡터 b 에서의 마지막 플래그가 1과 동일한 경우 동등하다. 이 상황에서, 말단 포지션 코딩은 바이패스될 수 있다. 예를 들어, $b = [100000000001]$ 이면, 런-길이 시퀀스는 '0-10-0'이다. '0-10'의 런-길이 시퀀스 (또는 위에서 설명된 예약된 런-길이 규칙들에 따른 시그널링된 값 시퀀스 0-11)은 이진 예측 벡터 b 를 복원하는데 충분할 수도 있다. 덧붙여, 이진 예측 벡터 b 가, 팔레트 엔트리들이 예측되지 않는다는 것을 의미하는, 영 벡터 (zero vector) 임을 나타내기 위해 임의의 런들을 시그널링하는 일 없이 말단 포지션이 시그널링될 수도 있다.
- [0161] 몇몇 예들에서, 남아있는 이진 플래그들이 0임을 시그널링하기 위해 고정된 값 L 을 사용하는 대신, L 및 이진 예측자 벡터의 사이즈 중 최소가 말단 포지션을 나타내는데 사용될 수도 있다. 이는 런의 값이 항상 이진 예측자 벡터의 사이즈보다 작기 때문이다.
- [0162] 도 7은 비디오 디코더 (30)의 팔레트 기반 디코딩 부 (165)의 일 예를 도시하는 블록도이다. 팔레트 기반 디코딩 부 (165)는 도 4의 팔레트 기반 인코딩 부 (122)에 대한 역 방식으로 수행하도록 구성될 수도 있다. 팔레트 기반 디코딩 부 (165)는 현재 블록에서의 각각의 화소에 대해, 팔레트에 대한 엔트리들이 현재 블록에서의 화소들에 대해 사용될 것인지의 여부를 나타내는 맵 (312)을 수신하도록 구성될 수도 있다. 덧붙여서, 맵 (312)은 무슨 팔레트 엔트리들이 주어진 화소에 대해 사용될 것인지를 추가로 나타낼 수도 있다. 맵 부 (302)는 디코딩된 비디오 데이터 (314)를 생성하기 위해 팔레트 생성 부 (304)에 의해 생성된 팔레트 및 맵 (312)을 사용하여 비디오 데이터의 현재 블록을 디코딩할 수도 있다.
- [0163] 본 개시물의 기법들에 따라, 팔레트 기반 디코딩 부 (165)는 인코딩된 이진 예측 벡터 (316)를 또한 수신할 수도 있다. 위에서 논의된 바와 같이, 이진 예측 벡터 (316)는 이진 예측 벡터에서의 0 값들의 런을 나타

내는 런-길이 시퀀스를 인코딩하는 런-길이 코딩 기법을 사용하여 인코딩될 수도 있다. 이진 예측 벡터 압축해제 부 (306) 는 도 4 내지 도 6을 참조하여 위에서 설명된 런-길이 코딩 기법들의 임의의 조합을 사용하여 인코딩된 이진 예측 벡터를 디코딩하도록 구성될 수도 있다. 일단 이진 예측 벡터가 이진 예측 벡터 압축해제 부 (306) 에 의해 회복되면, 팔레트 생성 부 (304) 는 비디오 데이터의 현재 블록에 대한 팔레트를 이진 예측 벡터와 예측자 팔레트 버퍼 (310) 에 저장된 이전에 사용된 팔레트 엔트리들에 기초하여 생성할 수도 있다.

팔레트 기반 디코딩 부 (165) 는 팔레트 기반 인코딩 부 (122) 가 이전에 사용된 팔레트 엔트리들을 예측자 팔레트 버퍼 (210) 에 저장했던 것과 동일한 방식으로, 이전에 사용된 팔레트 엔트리들을 예측자 팔레트 버퍼 (310) 에 저장하도록 구성될 수도 있다.

[0164] 도 8은 본 개시물의 일 예의 인코딩 방법을 도시하는 흐름도이다. 도 8의 기법들은 팔레트 기반 인코딩 부 (122) 를 포함하는 비디오 인코더 (20) 의 하나 이상의 구조적 컴포넌트들에 의해 구현될 수도 있다.

[0165] 본 개시물의 하나의 예에서, 팔레트 기반 인코딩 부 (122) 는 비디오 데이터의 현재 블록에 대한 팔레트를 생성하고 (800), 비디오 데이터의 현재 블록에 대한 팔레트에 대한 이진 예측 벡터를 생성하도록 (802) 구성될 수도 있다. 이진 예측 벡터는 이전에 사용된 팔레트 엔트리들이 비디오 데이터의 현재 블록에 대한 팔레트에 재사용되는지의 여부를 나타내는 엔트리들을 포함한다. 팔레트 기반 인코딩 부 (122) 는 또한, 런-길이 인코딩 기법을 사용하여 이진 예측 벡터를 인코딩하고 (804), 그 팔레트를 사용하여 비디오 데이터의 현재 블록을 인코딩하도록 (806) 구성될 수도 있다. 하나의 예에서, 런-길이 인코딩 기법은 영들의 런-길이를 코딩하는 것을 포함한다.

[0166] 본 개시물의 하나의 예에서, 팔레트 기반 인코딩 부 (122) 는 지수-콜롬 인코딩 기법을 사용하여 이진 예측 벡터를 인코딩하도록 구성될 수도 있다. 하나의 예에서, 지수-콜롬 인코딩 기법은 0차 지수-콜롬 인코딩 기법이다.

[0167] 본 개시물의 다른 예에서, 팔레트 기반 인코딩 부 (122) 는 런-길이 인코딩 기법과 예약된 런-길이 값 L 을 사용하여 이진 예측 벡터를 인코딩하도록 구성될 수도 있는데, 예약된 런-길이 값 L 은 이진 예측 벡터의 말단-포지션을 나타낸다. 하나의 예에서, 예약된 런-길이 값 L 은 1이다.

[0168] 본 개시물의 다른 예에서, 팔레트 기반 인코딩 부 (122) 는 런-길이 인코딩 기법, 예약된 런-길이 값 L , 및 최대 팔레트 사이즈를 사용하여 이진 예측 벡터를 인코딩하도록 구성될 수도 있다. 예약된 런-길이 값 L 은 이진 예측 벡터의 말단-포지션을 나타낸다. 이 예에서, 예약된 런-길이 값 L 은, 이전에 사용된 팔레트 엔트리들이 비디오 데이터의 현재 블록에 대한 팔레트에 재사용된다는 것을 나타내는 이진 예측 벡터에서의 엔트리들의 총 수가 최대 팔레트 사이즈와 동일하다면 사용되지 않는다.

[0169] 본 개시물의 다른 예에서, 팔레트 기반 인코딩 부 (122) 는 런-길이 인코딩 기법과 예약된 런-길이 값 L 을 사용하여 이진 예측 벡터를 인코딩하도록 구성될 수도 있는데, 예약된 런-길이 값 L 은 이진 예측 벡터의 말단-포지션을 나타낸다. 이 예에서, 예약된 런-길이 값 L 은, 인코딩된 이진 예측 벡터에서의 마지막 런-길이가 영의 런-길이를 나타내지 않는다면 사용되지 않는다.

[0170] 본 개시물의 다른 예에서, 팔레트 기반 인코딩 부 (122) 는 비디오 데이터의 하나 이상의 이전에 인코딩된 블록들에 대한 이전에 사용된 팔레트 엔트리들을 버퍼에 저장하도록 구성될 수도 있다. 팔레트 기반 인코딩 부 (122) 는 또한, 버퍼에 저장된 이전에 사용된 팔레트 엔트리들의 중복 엔트리들을 제거하도록 구성될 수도 있다. 본 개시물의 다른 예에서, 비디오 데이터의 하나 이상의 이전에 인코딩된 블록들에 대한 이전에 사용된 팔레트 엔트리들은 비디오 데이터의 현재 블록 상측의 화소들의 라인에 대한 이전에 사용된 팔레트 엔트리들과 비디오 데이터의 현재 블록 좌측의 화소들의 라인에 대한 이전에 사용된 팔레트 엔트리들을 포함한다.

[0171] 도 9는 본 개시물의 일 예의 디코딩 방법을 도시하는 흐름도이다. 도 9의 기법들은 팔레트 기반 디코딩 부 (165) 를 포함하는 비디오 디코더 (30) 의 하나 이상의 구조적 컴포넌트들에 의해 구현될 수도 있다.

[0172] 본 개시물의 하나의 예에서, 팔레트 기반 디코딩 부 (165) 는 비디오 데이터의 현재 블록에 대한 인코딩된 이진 예측 벡터를 수신하고 (900), 런-길이 디코딩 기법을 사용하여 인코딩된 이진 예측 벡터를 디코딩하도록 (902) 구성될 수도 있다. 하나의 예에서, 런-길이 디코딩 기법은 영들의 런-길이를 코딩하는 것을 포함한다. 팔레트 기반 디코딩 부 (165) 는 또한 이진 예측 벡터에 기초하여 비디오 데이터의 현재 블록에 대한 팔레트를 생성하도록 (904) 구성될 수도 있다. 이진 예측 벡터는 이전에 사용된 팔레트 엔트리들이 비디오 데이터의 현재 블록에 대한 팔레트에 재사용되는지의 여부를 나타내는 엔트리들을 포함한다. 팔레트 기반 디코딩 부 (165) 는 또한 그 팔레트를 사용하여 비디오 데이터의 현재 블록을 디코딩하도록 (906) 구성될 수도 있다.

- [0173] 본 개시물의 다른 예에서, 팔레트 기반 디코딩 부 (165) 는 또한 지수-골롬 디코딩 기법을 사용하여 인코딩된 이진 예측 벡터를 디코딩하도록 구성될 수도 있다. 하나의 예에서, 지수-골롬 디코딩 기법은 0차 지수-골롬 디코딩 기법이다.
- [0174] 본 개시물의 다른 예에서, 팔레트 기반 디코딩 부 (165) 는 또한, 런-길이 디코딩 기법과 예약된 런-길이 값 L 을 사용하여 인코딩된 이진 예측 벡터를 디코딩하도록 구성될 수도 있는데, 예약된 런-길이 값 L 은 이진 예측 벡터의 말단-포지션을 나타낸다. 하나의 예에서, 예약된 런-길이 값 L 은 1이다.
- [0175] 본 개시물의 다른 예에서, 팔레트 기반 디코딩 부 (165) 는 또한 런-길이 디코딩 기법, 예약된 런-길이 값 L, 및 최대 팔레트 사이즈를 사용하여 인코딩된 이진 예측 벡터를 디코딩하도록 구성될 수도 있다. 예약된 런-길이 값 L 은 이진 예측 벡터의 말단-포지션을 나타낸다. 예약된 런-길이 값 L 은, 이전에 사용된 팔레트 엔트리들이 비디오 데이터의 현재 블록에 대한 팔레트에 재사용된다는 것을 나타내는 이진 예측 벡터에서의 엔트리들의 총 수가 최대 팔레트 사이즈와 동일하다면 사용되지 않는다.
- [0176] 본 개시물의 다른 예에서, 팔레트 기반 디코딩 부 (165) 는 또한, 런-길이 디코딩 기법과 예약된 런-길이 값 L 을 사용하여 인코딩된 이진 예측 벡터를 디코딩하도록 구성될 수도 있는데, 예약된 런-길이 값 L 은 이진 예측 벡터의 말단-포지션을 나타낸다. 예약된 런-길이 값 L 은, 인코딩된 이진 예측 벡터에서의 마지막 런-길이가 영의 런-길이를 나타내지 않는다면 사용되지 않는다.
- [0177] 본 개시물의 다른 예에서, 팔레트 기반 디코딩 부 (165) 는 또한, 버퍼로부터, 이진 예측 벡터에 의해 팔레트에 대해 재사용되고 있는 것으로서 나타내어지는 이전에 사용된 팔레트 엔트리들을 팔레트 속에 복사하고, 팔레트에 복사된 이전에 사용된 팔레트 엔트리들의 수가 최대 팔레트 사이즈보다 작은 경우, 추가적인 팔레트 엔트리들을 수신하도록 구성될 수도 있다.
- [0178] 본 개시물의 다른 예에서, 팔레트 기반 디코딩 부 (165) 는 또한 비디오 데이터의 하나 이상의 이전에 디코딩된 블록들에 대한 이전에 사용된 팔레트 엔트리들을 버퍼에 저장하도록 구성될 수도 있다. 팔레트 기반 디코딩 부 (165) 는 또한, 버퍼에 저장된 이전에 사용된 팔레트 엔트리들의 중복 엔트리들을 제거하도록 구성될 수도 있다. 하나의 예에서, 비디오 데이터의 하나 이상의 이전에 디코딩된 블록들에 대한 이전에 사용된 팔레트 엔트리들은 비디오 데이터의 현재 블록 상측의 화소들의 라인에 대한 이전에 사용된 팔레트 엔트리들과 비디오 데이터의 현재 블록 좌측의 화소들의 라인에 대한 이전에 사용된 팔레트 엔트리들을 포함한다.
- [0179] 다음의 섹션들은 본 개시물의 추가적인 예의 기법들을 설명한다. 제 1 예에서, 비디오 인코더 (20) 는 골롬-라이스 코드, 지수-골롬 코드, 또는 절단된-라이스 코드, 또는 그것들의 조합을 사용하여 이진 예측 벡터 b 에서의 일들의 총 수 ('1의 수'), 즉, 예측자 팔레트 버퍼 (210)로부터의 재사용된 팔레트 엔트리들의 수를 먼저 시그널링하도록 구성될 수도 있다. 그 다음에, 일단 디코딩된 영이 아닌 이진 예측 플래그들의 수가 시그널링된 '1의 수'에 도달하면, 비디오 디코더 (30) 는 이진 예측 벡터 b 에서의 남아있는 플래그들이 0이고 이를 플래그들의 시그널링은 비디오 인코더 (20) 에 의해 바이패스되었다고 결정할 수도 있다. 다르게 말하면, 이 제 1 예에서, 위에서 설명된 말단 포지션 코딩 절차는 바이패스될 수 있다. 이 경우, 실제 런-길이들은 시그널링된다. 이러한 일 예에서, 런-길이 값들에 대한 상향 조정은 필요하지 않다.
- [0180] 제 2 예에서는, 위에서 설명된 바와 같이, 이진 예측 벡터에서의 영 값들의 런-길이만이 코드이다. 대체 기법으로서, 런-길이 코딩은 이진 예측 벡터에서의 0 값들의 런 길이들 및/또는 이진 예측 벡터에서의 1 값들의 런 길이들 양쪽 모두를 포함(예컨대, 허용) 할 수 있다. 초기 런은 '1의 런 길이' (또는 '0의 런 길이') 인 것으로 정의될 수 있다. 그 다음에, 비디오 인코더 (20) 는 영들의 런들 및 일들의 런들의 코딩을 교번 할 수도 있다.
- [0181] 예를 들어, 위에서와 동일한 예 (즉, $b = [110010001000]$) 를 사용하면 그리고 첫 번째 런이 '1의 런 길이'라고 가정하면, $b = [110010001000]$ 은 다음으로서 표현될 수 있다:
- [0182] '런=2'-'런=2'-'런=1'-'런=3'-'런=1'-'런=4'
- [0183] 그러나, '0 런' 다음에 '1 런'이 뒤따라야만 하고 반대의 경우도 마찬가지라고 알려져 있고; 그러므로, 위의 표현은 다음으로서 단순화될 수 있으며:
- [0184] '런=2'-'런=1'-'런=0'-'런=2'-'런=0'-'런=3'
- [0185] 이는 '210203'인 런-길이 시퀀스로서 표현될 수 있다. 골롬-라이스 코딩, 지수-골롬 코딩, 또는 절단된-라이스 코딩이 이 시퀀스를 코딩하는데 사용될 수 있다. '0 런'과 '1 런'은 골롬-라이스 코드, 지수-골롬 코

드, 또는 절단된-라이스 코드의 상이한 파라미터들을 가질 수도 있다. 정규 빈들이 '0 런'과 '1 런'을 인코딩하는데 사용된다면 그 '0 런'과 '1 런'에 대해 상이한 콘텍스트들이 사용될 수 있을 것이다. 본 개시물에서의 다른 곳에서 설명되는 말단 포지션 코딩 기법들은 또한 이 예에서 또한 적용될 수도 있다.

[0186] 덧붙여, 비디오 인코더 (20)는 시그널링된 '1' 엘리먼트들의 수가 최대 가능 팔레트 사이즈일 수도 있는 최대 가능 수에 도달하는 경우 런-길이 시그널링을 정지시키도록 구성될 수도 있다.

[0187] 제 3 예에서, 이진 예측 벡터 b 에서의 첫 번째 엘리먼트, 즉 b_0 는, 비디오 디코더 (30)가 첫 번째 런이 '0 런'인지 또는 '1 런'인지를 디코딩된 b_0 값에 기초하여 결정할 수도 있도록 직접적으로 시그널링될 수도 있다.

[0188] 제 4 예에서, 비디오 인코더 (20)는 이진 예측 벡터에서의 처음 M 개의 플래그들, 즉, $[b_0, b_1, \dots, b_{M-1}]$ 을 제 1 코딩 기법을 사용하여 인코딩하도록 구성될 수도 있고 나머지, 즉, $[b_M, b_{M+1}, \dots, b_{N-1}]$ 은 제 2 코딩 기법을 사용하여 코딩된다. 제 2 코딩 기법은 위에서 명시된 런-길이 기반 방법이다. 하나의 예에서, $M = 4$ 이다. 제 1 코딩 기법은 임의의 적합한 기준의 코딩 기법일 수도 있다. 일부 예들에서, 제 1 코딩 기법은 비-런 길이 코딩 기반 방법일 수도 있다. 다른 예들에서, 제 1 및 제 2 코딩 기법들은 둘 다가 런-길이 기반 코딩 기법들일 수도 있지만, 제 1 코딩 기법과 제 2 코딩 기법은 서로 상이할 수도 있다.

[0189] 이진 엘리먼트들 양쪽 모두에 대한 런-길이 기반 시그널링의 일 예에서는, 위에서 언급했듯이, 비디오 인코더 (20)는 이진 예측 벡터 b 에 대해 '0' 또는 '1'에 대한 런-길이 m 를 시그널링하도록 구성될 수도 있다. 그러나, 그 둘은 결합될 수도 있고 여기서 런-길이는 이진 엘리먼트들 양쪽 모두에 대해 시그널링된다. 이는 다음 단계들에 의해 달성될 수도 있다: (1) 처음에, bin 값, 예를 들어, $bin = 0$ 이 선택되면; (2) bin 엘리먼트에 대한 런-길이를 시그널링하며, (3) 빈이 1과 동일하면, 1과 동일한 bin 이 이진 벡터에서의 마지막 '1' 엘리먼트인지의 여부를 나타내는 마지막 플래그를 시그널링한다. 예를 들어, 마지막이 1과 동일한 것은 그것이 마지막 '1'임을 나타내고, 그렇지 않으면 0과 동일하다. 마지막이 1과 동일한 것은, 이진 벡터 엘리먼트들을 시그널링하는 것을 정지하고 종료하며; (4) 빈 값은 스왑되며, 즉, $bin = 1 - bin$ 이고 위의 단계 (2)로 진행한다.

[0190] 첫 번째 bin 값이 0인 것으로 선택되면, 이진 벡터가 영인 경우를 나타내는 것이 필요할 수도 있다. 본 개시물은 이진 벡터가 영인지의 여부를 나타내는 플래그를 $zero_flag$ 로서 지칭할 수도 있다. 예를 들어, 런-길이 시그널링의 처음에서의 별도의 플래그가 사용될 수도 있다. 별도의 플래그는 이진 벡터가 영임을 나타내는 1과 동일할 수도 있다. 이 경우, $zero_flag$ 뒤에는 런-길이 시그널링이 없다. $zero_flag$ 가 0이면, 런-길이 값들은 시그널링된다. 첫 번째 bin 이 1과 동일한 것으로 선택되면, 이 $zero_flag$ 를 시그널링할 필요가 없을 수도 있는데, 1과 동일한 bin 은 시그널링의 말단을 나타내는 $last_flag$ 를 갖기 때문이다.

[0191] 일 예에서, 위에서 논의된 기본 알고리즘은 다음 리스트에서의 기법들 중 하나 이상을 사용하여 추가로 개선될 수도 있다. 리스트에서의 각각의 아이템이 이 리스트에 후속하는 아래의 설명에서 "개선된 아이템"이라고 지칭될 수도 있다:

[0192] 1. bin 선택은 0 또는 1인 것으로 고정될, 또는 적응적일 (즉, 다양한 블록들에 대해 상이할) 수도 있다. 예를 들어, 어떤 엘리먼트가 가장 빈번한지 또는 이전에 코딩된 블록들에서 어떤 첫 번째 엘리먼트가 가장 빈번한지를 식별함으로써 bin 은 이전 블록들에 기초하여 선정될 (즉, 선택될) 수 있다. 대안으로, bin 은 모든 블록에 대해 시그널링될 수 있거나, 또는 블록들의 그룹에 대해 (예컨대, 각각의 타일에 대해 또는 슬라이스 헤더에서 시그널링될 수 있다. 대안으로, bin 은 비디오 시퀀스에서의 픽처마다, 예를 들어, 픽처 파라미터 세트 (PPS), 시퀀스 파라미터 세트 (SPS), 비디오 파라미터 세트 (VPS), 또는 다른 경우 중 적어도 하나에서 시그널링될 수 있다.

[0193] 2. 마지막 플래그는, 이 마지막 플래그가 시그널링되는 이진 벡터 엘리먼트 '1'이 이진 벡터 b 에서의 마지막 엘리먼트이면 시그널링되지 않을 수도 있다.

[0194] 3. 이진 벡터에서의 첫 번째 엘리먼트를 프로세싱한 후, 시그널링된 런-길이 값은 실제 런-길이 값 빼기 1일 수 있는데, 그 엘리먼트가 마지막 것이 아니면, 그것에는 적어도 하나의 다른 이진 엘리먼트가 뒤따른다는 것이 알려져 있기 때문이다.

[0195] 4. 이진 벡터에서의 마지막 '1' 엘리먼트에 대한 런-길이 값은, 이진 벡터에서의 마지막 '1' 엘리먼트에 대한 런-길이 값에 '0' 엘리먼트가 선행한다면 시그널링되지 않을 수도 있는데, '0' 런-길이 값이 이진 벡터에서의 마지막 엘리먼트 앞에서 정지된다면 이 엘리먼트는 '1'이고 그것은 그 벡터에서의 마지막 '1' 엘리먼트라

는 것이 알려져 있기 때문이다.

[0196] 5. 대안으로, zero_flag 시그널링에 대해, 이진 벡터에서의 마지막 '1' 엘리먼트에 대한 런-길이 값은 이 것에 대해 특정 값이 배정되는 첫 번째 런-길이 값에 포함될 수도 있다. 예를 들어, 첫 번째 런-길이 값이 영이면, 이진 벡터는 영이고 그 후에 다른 시그널링은 없다. 영이 아닌 이진 벡터에 대해, 첫 번째 시그널링되는 런-길이 값은 모호성을 피하기 위해 그 특정 값, 이 예에서는 0 다음에 1만큼 증가되는 것이 필요하다.

[0197] 6. zero_flag와 last_flag는 콘텍스트 적응 이진 산술 코딩 (CABAC) 콘텍스트 코딩될 수도 있으며, 예를 들어, 별도의 콘텍스트가 각각의 플래그에 대해 배정될 수 있다. 콘텍스트들은 블록 사이즈, 슬라이스 유형 등에 의존적일 수 있다.

[0198] 7. 런-길이 값들은 런-길이 값이 이진 벡터 사이즈보다 더 클 수 없다는 것을 고려하여 다양한 이진화 방법들, 이를테면 단항 코드, 골롬-라이스 코드 또는 지수-골롬, 또는 그 코드들의 절단된 버전들을 사용하여 코딩될 수 있다.

[0199] 8. 런-길이 시그널링은 시그널링된 '1' 엘리먼트들의 수가 최대 가능 수에 도달한 경우 정지될 수 있는데, 최대 가능 수는 최대 가능 팔레트 사이즈를 표현할 수 있다.

[0200] 이진 예측 벡터 $b = \{11001110100001\}$ 를 갖는 일 예에 대해, 위에서 설명된 기본 알고리즘은 다음을 생성한다: [0]0-2(0)-2-3(0)-1-1(0)-4-1 (1). 처음에서, bin이 0으로 설정되고, 첫 번째 런-길이는 그 빈에 시그널링되고 그것은 이 예에서 0이며, 그러면 모든 '-'에서 bin은 1-bin으로 스왑된다. 시그널링된 마지막 플래그는 팔호들 내에서 나타내어진다. 객석팔호들 내에 0-플래그가 있다. 일 예에 대해, 다음은 개선된 알고리즘의 결과이다: 1-1(0)-1-2(0)-0-0(0)-3-0(1).

[0201] 위의 아이템 #5로 인한 런-길이 값 1을 반영하는 제 1 엘리먼트 프로세싱 후, 다른 런-길이 값들은 위의 개선 아이템 #3에서 언급된 바와 같이 1만큼 감소된다. 취소선은 위의 #2에서 언급된 벡터에서의 마지막 '1'에 대한 마지막 플래그의 시그널링은 없다는 것과, 위의 #4에서 제안된 마지막 '1' 엘리먼트에 대한 런-길이 값 시그널링은 없다는 것을 나타낸다.

[0202] 다른 예에 대해: $b = \{11001110100011\}$ 와 일 예의 개선된 알고리즘은 다음을 초래한다: 1-1(0)-1-2(0)-0-0(0)-3-1(1).

[0203] 일부 예의 방법들은 팔레트 모드 인덱스 블록 코딩에서 상측 복사 런, 좌측 복사 런, 또는 둘 다의 길이를 코딩하기 위해 지수-골롬 코드를 사용할 수도 있다. 예를 들어, 런 길이를 코딩하기 위해 지수-골롬 코드를 사용하는 방법이 팔레트 모드 인덱스 블록 코딩에서 상측 복사 런, 좌측 복사 런, 또는 둘 다의 길이를 코딩하기 위해 지수-골롬 코드를 또한 적용할 수도 있다. 다른 예에서, "0보다 더 큼", "1보다 더 큼", 및 "2보다 더 큼" 플래그들을 시그널링한 후, 2차 골롬-라이스 코드가 (예컨대, 런 길이 - 3) ≥ 0 이면 남아있는 값들 (예컨대, 런 길이 - 3) 을 코딩하는데 사용될 수도 있다.

[0204] 일부 예들에서, "0보다 더 큼", "1보다 더 큼", 및 "2보다 더 큼" 플래그들은 런 길이가 "0보다 더 큼", "1보다 더 큼", 또는 "2보다 더 큼"을 각각 나타내기 위해 사용될 수도 있다. 런-길이가 런-길이 코딩된 이진 예측 벡터를 사용하여 코딩될 수도 있다.

[0205] 본 개시물의 일부 추가적인 예들에 따르면, 비디오 코더 (예컨대, 비디오 인코더 (20) 또는 비디오 디코더 (30)) 가 하나 이상의 새로운 팔레트 예측자들을 결정할 수도 있다. 본 개시물의 이들 추가적인 예들에 따르면, 비디오 코더는 팔레트 예측자들을 시그널링하기 위한 새로운 불리언 벡터를 결정할 수도 있다. 이들 및 다른 기법들은 아래의 도 10에 관하여 더 자세히 설명된다.

[0206] 도 10은 본 개시물의 기법들과 일치하게, 비디오 데이터를 코딩하기 위한 팔레트를 결정하는 일 예를 도시하는 개념도이다. 도 10의 예는 제 1 팔레트들 (184) 에 연관되는 팔레트 모드 (PAL) 를 사용하여 코딩되는 제 1 코딩 유닛 (CU) (180) 과 제 2 팔레트들 (192) 에 연관되는 제 2 CU (188) 를 갖는 핵심 (178) 를 포함한다.

아래에서 더욱 상세히 그리고 본 개시물의 기법들에 따라 설명되는 바와 같이, 제 2 팔레트들 (192) 은 제 1 팔레트들 (184) 에 기초한다. 핵심 (178) 는 인트라 예측 코딩 모드로 코딩된 블록 (196) 과 인터 예측 코딩 모드로 코딩되어 있는 블록 (200) 을 또한 포함한다.

[0207] 도 10의 기법들은 설명의 목적으로 비디오 인코더 (20) (도 1 및 도 2) 와 비디오 디코더 (30) (도 1 및 도 3) 에 의한 수행의 맥락에서 그리고 HEVC 비디오 코딩 표준에 관해 설명된다. 그러나, 본 개시물의 기법들은

이런 식으로 제한되지 않고, 다른 비디오 코딩 프로세스들 및/또는 표준들에서 다른 비디오 코딩 프로세서들 및/또는 디바이스들에 의해 적용될 수도 있다는 것이 이해되어야 한다.

[0208] 대체로, 팔레트가 현재 코딩되고 있는 CU, 즉 도 10의 예에서의 CU (188)에 대해 우세하거나 및/또는 대표하는 다수의 화소 값들에 관련된다. 제 1 팔레트들 (184)과 제 2 팔레트들 (192)은 다수의 팔레트들을 포함하는 것으로서 도시되어 있다. 일부 예들에서, 본 개시물의 양태들에 따르면, 비디오 코더 (이를테면 비디오 인코더 (20) 또는 비디오 디코더 (30))가 CU의 각각의 컬러 성분에 대해 팔레트들을 따로따로 코딩할 수도 있다. 예를 들어, 비디오 인코더 (20)는 CU의 루마 (Y) 성분에 대한 팔레트, CU의 크로마 (U) 성분에 대한 다른 팔레트, 및 CU의 크로마 (V) 성분에 대한 또 다른 팔레트를 인코딩할 수도 있다. 이 예에서, Y 팔레트의 엔트리들은 CU의 화소들의 Y 값들을 표현할 수도 있으며, U 팔레트의 엔트리들은 CU의 화소들의 U 값들을 표현할 수도 있고, V 팔레트의 엔트리들은 CU의 화소들의 V 값들을 표현할 수도 있다.

[0209] 다른 예들에서, 비디오 인코더 (20)는 CU의 모든 컬러 성분들에 대해 단일 팔레트를 인코딩할 수도 있다. 이 예에서, 비디오 인코더 (20)는 Y_i , U_i , 및 V_i 를 포함하는 3중 값인 i 번째 엔트리를 갖는 팔레트를 인코딩할 수도 있다. 이 경우, 팔레트는 화소들의 성분들의 각각에 대한 값들을 포함한다. 따라서, 다수의 개개의 팔레트들을 갖는 팔레트들의 세트로서의 팔레트들 (184 및 192)의 표현은 단지 하나의 예이고 제한하려는 의도는 아니다.

[0210] 도 10의 예에서, 제 1 팔레트들 (184)은 엔트리 인덱스 값 1, 엔트리 인덱스 값 2, 및 엔트리 인덱스 값 3을 각각 갖는 세 개의 엔트리들 (202~206)을 포함한다. 엔트리들 (202~206)은 인덱스 값들을 화소 값 A, 화소 값 B, 및 화소 값 C를 각각 포함하는 화소 값들에 관련시킨다. 본원에서 설명된 바와 같이, 제 1 CU (180)의 실제 화소 값을 코딩하는 것이 아니라, 비디오 코더 (이를테면 비디오 인코더 (20) 또는 비디오 디코더 (30))가 인덱스들 (1~3)을 사용하여 블록의 화소들을 코딩하기 위해 팔레트 기반 코딩을 사용할 수도 있다. 다시 말하면, 제 1 CU (180)의 각각의 화소 포지션에 대해, 비디오 인코더 (20)는 화소에 대한 인덱스 값을 인코딩할 수도 있으며, 여기서 인덱스 값은 제 1 팔레트들 (184) 중 하나 이상의 제 1 팔레트들에서의 화소 값에 연관된다. 비디오 디코더 (30)는 비트스트림으로부터 인덱스 값들을 획득하고 그 인덱스 값을 및 하나 이상의 제 1 팔레트들 (184)을 사용하여 화소 값들을 복원할 수도 있다. 따라서, 제 1 팔레트들 (184)은 팔레트 기반 디코딩에서 비디오 디코더 (30)에 의한 사용을 위해 비디오 인코더 (20)에 의해 인코딩된 비디오 데이터 비트스트림에서 송신된다.

[0211] 일부 예들에서, 비디오 인코더 (20)와 비디오 디코더 (30)는 제 1 팔레트들 (184)에 기초하여 제 2 팔레트들 (192)을 결정할 수도 있다. 예를 들어, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는 예측 팔레트들, 이 예에서는 제 1 팔레트들 (184)이 결정되는 하나 이상의 블록들의 위치를 찾아낼 수도 있다. 일부 예들, 이를테면 도 10에 도시된 예에서, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는, 제 2 CU (188)에 대한 예측 팔레트를 결정하는 경우, 좌측 이웃 CU (제 1 CU (180))와 같은 이전에 코딩된 CU의 위치를 찾아낼 수도 있다.

[0212] 도 10의 예에서, 제 2 팔레트들 (192)은 엔트리 인덱스 값 1, 엔트리 인덱스 값 2, 및 엔트리 인덱스 값 3을 각각 갖는 세 개의 엔트리들 (208~212)을 포함한다. 엔트리들 (208~212)은 인덱스 값들을 화소 값 A, 화소 값 B, 및 화소 값 D를 각각 포함하는 화소 값들에 관련시킨다. 이 예에서, 비디오 인코더 (20)는 제 1 팔레트들 (184) 중 어떤 엔트리들이 제 2 팔레트들 (192)에 포함되는지를 나타내는 하나 이상의 신팩스 엘리먼트들을 코딩할 수도 있다. 도 10의 예에서, 하나 이상의 신팩스 엘리먼트들은 벡터 (216) (예컨대, 이진 예측 벡터)로서 예시된다. 벡터 (216)는 다수의 연관된 빈들 (또는 비트들)을 가지며; 각각의 빈은 그 빈에 연관된 팔레트 예측자가 현재 팔레트의 엔트리를 예측하기 위해 사용되는지의 여부를 나타낸다. 예를 들어, 벡터 (216)는 제 1 팔레트들 (184)의 처음 두 개의 엔트리들 (202 및 204)이 제 2 팔레트들 (192)에 포함된다는 것을 나타내는 반면 (벡터 (216)에서의 "1"의 값), 제 1 팔레트들 (184)의 제 3 엔트리는 제 2 팔레트들 (192)에 포함되지 않는다 (벡터 (216)에서의 "0"의 값). 도 10의 예에서, 벡터는 불리언 벡터이다.

[0213] 일부 예들에서, 비디오 인코더 (20)와 비디오 디코더 (30)는 팔레트 예측을 수행하는 경우 팔레트 예측자 리스트 (이는 팔레트 예측자 테이블이라고 또한 지칭될 수도 있음)를 결정할 수도 있다. 팔레트 예측자 리스트는 현재 블록을 코딩하기 위해 팔레트의 하나 이상의 엔트리들을 예측하는데 사용되는 하나 이상의 이웃 블록들의 팔레트들로부터의 엔트리들을 포함할 수도 있다. 비디오 인코더 (20)와 비디오 디코더 (30)는 그 리스트를 동일한 방식으로 구축할 수도 있다. 비디오 인코더 (20)와 비디오 디코더 (30)는 팔레트 예측자

리스트의 어떤 엔트리들이 현재 블록을 코딩하기 위해 팔레트에 포함될 것인지를 나타내기 위해 데이터 (이를테면 벡터 (216)) 를 코딩할 수도 있다.

[0214] 문서 『C. Gisquet, G. Laroche, and P. Onno, "AhG10: Palette predictor stuffing," JCTVC-Q0063』은 팔레트 예측자 리스트를 결정하는 하나의 예의 프로세스를 개시한다. 일부 예들에서, 위에서 언급했듯이, 비디오 인코더 (20) 또는 비디오 디코더 (30) 는 팔레트 예측자 리스트에서의 각각의 아이템이 현재 코딩되고 있는 블록에 대한 팔레트에서의 하나 이상의 엔트리들을 예측하기 위해 사용되는지 (또는 사용되지 않는지) 의 여부를 나타내기 위해 불리언 벡터 (이를테면 벡터 (216)) 를 사용할 수도 있다.

[0215] 일부 예들에서, 팔레트 예측자 리스트에서의 아이템들의 모두는 이전에 코딩된 팔레트 (예컨대, 이전에 코딩된 블록으로 코딩된 팔레트)로부터 도출된다. 그러나, 이러한 팔레트들은 현재 CU로부터 공간적으로 떨어져 있을 수도 있는데, 이는 팔레트 상관을 상대적으로 약하게 할 수도 있다. 대체로, 팔레트 예측자 테이블 (또는 리스트) 을 확장하는 것이 도움이 될 수도 있다 (예컨대, 더욱 정확한 예측자들을 제공할 수도 있으며, 이는 효율성 증대를 초래할 수도 있다). 그러나, 비교적 큰 팔레트 예측자 테이블 (또는 리스트) 을 결정 및 사용하는 것은 상대적으로 더 긴 불리언 벡터를 초래한다.

[0216] 다양한 예들에서, 본 개시물에서 설명되는 기법들은 팔레트-기반 코딩에서 팔레트들을 결정, 예측, 및/또는 시그널링하는 다양한 조합들을 위한 기법들을 포함할 수도 있다. 그 기법들은 팔레트 예측자들을 결정하는 것, 예측자들을 예측자들의 후보 리스트에 추가하는 것, 예측자들의 후보 리스트로부터 예측자들을 프루닝하는 것, 후보 예측자들의 사용의 표시를 코딩하는 것, 또는 본원에서 설명되는 임의의 다른 기법들의 임의의 조합을 포함할 수도 있다. 비록 특정 예들이 예시 및 명료함을 목적으로 개별적으로 설명될 수도 있지만, 본 개시물은 본원에서 설명되는 팔레트-기반 코딩을 위한 기법들의 임의의 조합을 생각해 낼 수도 있다.

[0217] 본 개시물의 일부 양태들에 따르면, 비디오 코더 (이를테면 비디오 인코더 (20) 또는 비디오 디코더 (30)) 가 하나 이상의 새로운 팔레트 예측자들을 결정할 수도 있다. 예를 들어, 본 개시물의 특정 기법들은 하나 이상의 공간적 팔레트 예측자들을 결정하는 것을 포함한다. 일 예에서, 현재 CU 상측의 CU ("상측 CU" 또는 "상부 CU"라고 지칭되고 도 10의 예에서 인트라 코딩된 CU (196) 로서 도시됨) 가 팔레트-기반 코딩을 사용한다면, 상부 CU로부터의 팔레트는 현재 CU (제 2 CU (188)) 에 대한 팔레트 예측자로서 효과적이다. 다시 말하면, 상부 CU의 엔트리들의 각각은 현재 팔레트를 예측하기 위한 팔레트 예측자 리스트 (예컨대, 이는 어떤 엔트리들이 예측자들로서 사용되는지를 나타내는 벡터에 연관될 수도 있음) 에 포함될 수도 있다.

[0218] 그러나, 상측 CU가 상이한 CTU의 부분이면, 예측을 위해 상측 CU로부터의 팔레트를 사용하는 것은 추가적인 저장을 요구할 수도 있다. 예를 들어, 인터 예측된 블록에 대해, 비디오 코더는 화소 값들을 복원하기 위해 메모리 (예컨대, 비디오 코더에 대한 외부 메모리) 에 액세스해야 할 수도 있으며, 이는 레이턴시를 초래할 수도 있다. 그러나, 비디오 코더는, 일부 예들에서, 인트라 예측된 블록에 연관됨으로써 참조 샘플들로서 표시되는 복원된 샘플들을 고려할 수도 있다. 이러한 복원된 샘플들은 비디오 인코더 (20) 및 비디오 디코더 (30) 둘 다에 국부적으로 저장되고 이용 가능할 수도 있다.

[0219] 일 예에서, 본 개시물의 양태들에 따르면, 참조 샘플들은 팔레트 예측자 리스트에 추가된다. 예를 들어, 공간적으로 이웃하는 블록의 화소 값들 (이것들은 팔레트 코딩 또는 인트라 코딩될 수도 있음) 은 (위에서 언급했듯이) 후보 팔레트 예측자들로서 식별되고 벡터에 연관될 수도 있다. 일부 예들에서, 참조 샘플들의 서브셋만이 사용될 수도 있다. 예를 들어, 팔레트 예측자 리스트가 특정 CU, 이를테면 좌측 이웃 CU (제 1 CU (180)) 로부터의 팔레트를 이미 포함한다면, 그 CU로부터의 샘플들은 팔레트 예측 리스트에 포함되지 않을 수도 있다. 더욱이, 일부 예들에서, 비디오 코더는 비디오 인코더 (20) 와 특히 비디오 디코더 (30) 에 대한 추가적인 복잡도를 대가로, 그 리스트에서의 중복된 팔레트 예측자들을 제거하기 위해 프루닝 프로세스를 적용할 수도 있다. 이들 새로운 팔레트 예측자들은 예측자 리스트의 처음 또는 그 리스트의 말단 중 어느 하나에 삽입될 수도 있다. 다른 예들에서, 새로운 팔레트 예측자들 (예컨대, 공간적 팔레트 예측자들) 은 예측자 리스트 사이즈, CU 사이즈 등과 같은 특정 규칙에 따라 그 리스트 속에 적응적으로 삽입될 수도 있다.

[0220] 다른 예들에서, 현재 CU 좌측의 열에서의 임의의 CU의 화소들과 같은 임의의 다른 복원된 화소들이 팔레트 예측자 리스트에 또한 추가될 수도 있다.

[0221] 이런 방식으로, 비디오 인코더 (20) 및/또는 비디오 디코더 (30) 는 비디오 데이터의 현재 블록에 대한 팔레트의 하나 이상의 엔트리들을 예측하기 위한 하나 이상의 공간적 팔레트 예측자들을 결정하는 것으로서, 하나 이상의 엔트리들 중 각각의 엔트리는 화소 값을 나타내는, 상기 하나 이상의 공간적 팔레트 예측자들을 결정하고,

하나 이상의 팔레트 예측자들 중 각각의 팔레트 예측자가 비디오 데이터의 현재 블록에 대한 팔레트의 엔트리들을 예측하는데 사용되는지의 여부를 나타내는 하나 이상의 십екс 엘리먼트들을 코딩할 수도 있다.

[0222] 본 개시물의 다른 양태들에 따르면, 비디오 코더 (이를테면 비디오 인코더 (20) 또는 비디오 디코더 (30)) 가 팔레트 예측자들을 시그널링하기 위한 새로운 불리언 벡터를 결정할 수도 있다. 일부 예들에서, 본 개시물의 기법들은 팔레트 예측을 위해 불리언 벡터를 인코딩하는데 사용될 수도 있다.

[0223] 예를 들어, 본 개시물의 양태들에 따르면, 비디오 인코더 (20) 또는 비디오 디코더 (30) 는 팔레트 예측을 위한 이진 트리를 결정할 수도 있다. 예시 목적을 위한 일 예에서, 팔레트 예측자들의 사용을 표현하는 0-1 불리언 벡터가 N에 의해 표시된다고 하자. 이 벡터를 효율적으로 인코딩하기 위해, 이진 트리가 사용된다. 구체적으로는, 벡터 N은 여러 인접한 지역들로 구획화되는데, 각각의 지역은 하나 또는 하나를 초과하는 벡터 엘리먼트들을 포함한다. 하나의 비트가 그 지역 내의 엘리먼트들의 모두가 영인지의 여부를 나타내기 위해 시그널링된다. 그 지역이 모두 영인 것은 아니면, 그 지역은 동일한 방도로 서브 지역들로 추가로 분리된다.

[0224] 구획화는 주어진 규칙들에 따라 정지될 수도 있다. 예를 들어, 하나의 규칙이 "지역 사이즈가 임계값보다 더 작다면, 파티션 프로세스는 정지한다"일 수도 있다. 구획화가 정지되는 경우, 지역 사이즈가 1보다 더 크고 지역이 영이 아닌 엘리먼트들을 포함한다면, 그 지역에서의 각각의 엘리먼트는 비트스트림에서 시그널링된다. 이 경우, 지역 사이즈가 X라고 가정하면, 첫 번째 X-1 엘리먼트들이 모두 영이면, 마지막 엘리먼트는 비트스트림에 포함되지 않을 수도 있도록 1이 되어야만 한다.

[0225] 다른 예에서, 구획화 프로세스가 정지되고 지역 사이즈가 1보다 더 크지만 제 2 임계값보다 작은 경우, 비디오 코더는 그 지역에서의 엘리먼트들의 모두가 영인지의 여부를 나타내기 위해 1 비트의 시그널링을 스킵할 수도 있다. 대신, 그 지역에서의 각각의 엘리먼트에 대해, 비디오 코더는 각각의 엘리먼트가 영인지 또는 1인지를 나타내기 위해 하나의 비트를 시그널링할 수도 있다.

[0226] 다른 예에서, 이진 트리 대신, 각각의 노드에서 두 개를 초과하는 분기들을 갖는 트리가 사용될 수도 있다. 또한, 일부 예들에서, 트리의 루트로부터 시작하는 처음의 여러 레벨들의 인코딩은 바이패스될 수도 있다. 그런 예들에서, 그 값들은 1인 것으로 암시적으로 가정될 수도 있다.

[0227] 본 개시물의 양태들에 따르면, 팔레트 예측자 리스트로부터의 마지막 영이 아닌 엔트리의 포지션 (즉, 불리언 벡터에서의 마지막 1) 은, 개개의 엔트리들이 0인지 또는 1인지를 시그널링하기 전에, 비트스트림에서 명시적으로 코딩될 수도 있다. 이 경우, 마지막 포지션 (을 포함한) 부터 불리언 벡터의 말단까지의 엘리먼트들은 비트스트림에서 시그널링되지 않는다. 다른 예에서, 예측을 위해 사용되는 첫 번째 팔레트 예측자의 포지션, 즉, 불리언 벡터에서의 첫 번째 1은, 비트스트림에서 또한 명시적으로 시그널링될 수도 있다.

[0228] 본 개시물의 양태들에 따르면, 팔레트 예측자 리스트로부터 사용되는 엔트리들의 수는 비트스트림에서 명시적으로 코딩될 수도 있다. 예를 들어, 본 개시물의 기법들은 위에서 논의된 불리언 벡터에서 그 일들 ("1들") 의 수를 시그널링하는 것을 포함한다. 일부 예들에서, 비디오 인코더 (20) 는 불리언 벡터에서 "1들"의 수 (*TotalOnes*) 를 초기에 시그널링할 수도 있다. 비디오 인코더 (20) 는 그 다음에 시그널링된 1들의 수가 *TotalOnes*와 동일하기까지 불리언 벡터에서의 각각의 아이템 (0 또는 1) 을 순차적으로 시그널링 (또는 일부 다른 방식으로 시그널링) 할 수도 있다.

[0229] 비슷하게, 비디오 디코더 (30) 는 수신된 불리언 벡터에 포함되어 있는 1들의 수의 표시를 먼저 디코딩할 수도 있다. 비디오 디코더 (30) 는 그 다음에 비디오 디코더 (30) 가 나타내어진 "1들"의 수에 매칭되는 벡터에서의 "1들"의 수를 디코딩하기까지, 수신된 벡터의 각각의 엘리먼트를 디코딩할 수도 있다. 일부 예들에서, (예컨대, 벡터의 엘리먼트들에 대한) 아이템-식 시그널링은 벡터의 처음부터 순차적일 수도 있다. 다른 예들에서, 벡터의 엘리먼트들은 일부 다른 스캐닝 순서에 따라 시그널링될 수도 있다. 또 다른 예들에서, 아이템-식 시그널링이 런-길이 코딩 기술을 사용할 수도 있다.

[0230] 일부 예들에서, 불리언 벡터에서의 1들의 수가 초기에 시그널링되는 (예컨대, *TotalOnes*가 불리언 벡터전에 비트스트림에서 시그널링되는) 경우, 불리언 벡터에서의 각각의 십екс 엘리먼트 (예컨대, 플래그) 는 CABAC를 사용하여 콘텍스트 코딩될 수도 있다. 일부 경우들에서, 다수의 콘텍스트들이 사용될 수도 있다. 일부 예들에서, 불리언 벡터의 십екс 엘리먼트들의 모두가 콘텍스트 코딩될 수도 있다. 다른 예들에서, 불리언 벡터의 십екс 엘리먼트들의 서브세트만이 콘텍스트 코딩될 수도 있다.

[0231] 본 개시물의 양태들에 따르면, 불리언 벡터의 십екс 엘리먼트에 대한 콘텍스트는 불리언 벡터 내의 십екс 엘

리먼트 (예컨대, 이진 플래그)의 포지션 및 초기에 시그널링되는 시그널링된 1들의 수 (*TotalOnes*)에 의존적일 수도 있다. 예시 목적을 위한 일 예에서, 만약 ((포지션 > 3) && ((포지션/2+1) > *TotalOnes*))이면 비디오 코더 (이를테면 비디오 인코더 (20) 또는 비디오 디코더 (30))는 제 1 콘텍스트를 사용할 수도 있으며; 그렇지 않으면 (조건이 참이 아닌 경우) 비디오 코더는 다른 콘텍스트를 사용할 수도 있는데, 여기서 포지션은 불리언 벡터에서의 코딩된 신택스 엘리먼트의 상대 포지션을 나타낸다. 이 경우, 비디오 코더는 불리언 벡터에서의 신택스 엘리먼트들 (예컨대, 이진 플래그들)을 코딩하기 위해 두 개의 콘텍스트들을 사용하고, 불리언 벡터에서의 특정 신택스 엘리먼트의 상대 포지션에 기초하여 불리언 벡터의 특정 신택스 엘리먼트에 대한 콘텍스트를 선택한다. 다른 예에서, 비디오 코더는 만약 조건 ((포지션 > 3) && ((포지션/4+1) >= *TotalOnes*))이 참이면 하나의 콘텍스트를 그리고 그 조건이 참이 아니면 다른 콘텍스트를 사용할 수도 있다. 콘텍스트들을 결정하기 위한 다른 조건들이 또한 사용될 수도 있다.

[0232]

위의 예들에서, 불리언 벡터의 특정 신택스 엘리먼트에 대한 CABAC 콘텍스트는 신택스 엘리먼트의 상대 포지션에 그리고 불리언 벡터에서의 1들의 총 수 (*TotalOnes*)에 의존한다. 다른 예들에서, 포지션 및 *TotalOnes*에 대한 의존성 대신, 비디오 코더는 코딩되고 있는 신택스 엘리먼트의 상대 포지션과 지금까지 시그널링된 1들의 총 수 (예컨대, 불리언 벡터에 코딩되어 있는 신택스 엘리먼트보다 앞에 발생하였던 1들의 총 수)에 기초하여 신택스 엘리먼트들에 대한 콘텍스트를 결정할 수도 있다. 또 다른 예들에서, 비디오 코더가 불리언 벡터의 신택스 엘리먼트들을 CABAC 코딩하기 위한 콘텍스트를 결정하기 위해 위에서 설명된 조건들의 임의의 조합을 사용할 수도 있다.

[0233]

본 개시물의 양태들에 따르면, 팔레트 사이즈가 인코더에 의해 시그널링되고 디코더에 의해 수신된다. 예를 들어, 팔레트에서의 엔트리들의 수가 팔레트의 사이즈를 나타내기 위해 시그널링될 수도 있다. 일부 예들에서, 비디오 인코더 (20)는 팔레트 사이즈 (*PLTSize*)를 초기에 시그널링할 수도 있다. 비디오 인코더 (20)는 그 다음에 시그널링된 1들의 수가 *PLTSize*와 동일하기까지 불리언 벡터에서 각각의 아이템 (0 또는 1)을 시그널링할 수도 있다. 비디오 디코더 (30)는 팔레트 사이즈의 표시를 수신할 수도 있고, 팔레트에서의 엔트리들의 수가 시그널링된 팔레트 사이즈에 도달하기까지 (예컨대, 벡터에 의해 식별된 팔레트 예측자들을 사용하여) 팔레트에 계속 추가할 수도 있다.

[0234]

일부 예들에서, 비디오 인코더 (20)는 팔레트 사이즈 (*PLTSize*)를 초기에 시그널링할 수도 있다. 비디오 인코더 (20)는 그 다음에 비예측된 팔레트 아이템들의 수 (*nonPredPLTSize*)를 시그널링할 수도 있다. 비디오 인코더 (20)는 그 다음에 시그널링된 "1들"의 수가 팔레트 사이즈와 비예측된 팔레트 아이템들의 수 간의 차이 (*PLTSize* - *nonPredPLTSize*)와 동일하기까지 불리언 벡터에서 각각의 아이템 (0 또는 1)을 시그널링할 수도 있다. 일부 예들에서, 비디오 디코더 (30)는 팔레트 사이즈의 표시와 팔레트 예측자를 사용하여 예측되지 않는 팔레트에서의 엔트리들의 수 (예컨대, 벡터의 "0들"의 수)를 수신할 수도 있다. 비디오 디코더 (30)는 어떤 팔레트 예측자들이 팔레트에 포함될 것인지를 나타내는 벡터 (예컨대, 이를테면 위에서 설명된 불리언 벡터)를 또한 수신할 수도 있다. 비디오 디코더 (30)는 그 다음에 "1들"의 수가 팔레트 사이즈와 비예측된 팔레트 아이템들의 수 간의 차이 (*PLTSize* - *nonPredPLTSize*)와 동일하기까지 그 벡터에서의 각각의 아이템을 디코딩할 수도 있다.

[0235]

이런 방식으로, 비디오 인코더 (20) 및/또는 비디오 디코더 (30)는, 하나의 예에서, 비디오 데이터의 현재 블록에 대한 팔레트의 하나 이상의 엔트리들을 예측하기 위한 하나 이상의 팔레트 예측자들을 결정하는 것으로서, 하나 이상의 엔트리들 중 각각의 엔트리는 화소 값을 나타내는, 상기 하나 이상의 팔레트 예측자들을 결정하고, 벡터의 각각의 빈이 하나 이상의 팔레트 예측자들의 각각의 팔레트 예측자가 비디오 데이터의 현재 블록에 대한 팔레트의 엔트리를 예측하는데 사용되는지의 여부를 나타내는 복수의 빈들을 갖는 벡터를 결정하며, 그 벡터를 하나 이상의 지역들로 구획화하고, 그 구획화에 기초하여 벡터의 빈들의 하나 이상의 값들을 결정할 수도 있다.

[0236]

비록 위에서 설명된 도 10의 기법들이 CU들 (HEVC)의 맥락에서 설명되지만, 그 기법들은 예측 유닛들 (PU들)에 또는 다른 비디오 코딩 프로세스들 및/또는 표준들에서 또한 적용될 수도 있다는 것이 이해되어야 한다.

[0237]

예에 의존하여, 상이한 시퀀스로 수행될 수도 있는 본원에서 설명된 기법들 중 임의의 것의 특정한 액트들 또는 이벤트들이 부가되거나, 병합되거나, 또는 다 함께 제외될 수도 있다 (예컨대, 모든 설명된 액트들 또는 이벤트들이 그 기법들의 실용화에 필요한 것은 아니다)는 것이 이해되어야 한다. 더구나, 특정 예들에서, 액트들 또는 이벤트들은 순차적으로라기 보다는, 예컨대, 다중 스레드식 프로세싱, 인터럽트 프로세싱, 또는 다수의 프로세서들을 통하여 동시에 수행될 수도 있다. 덧붙여서, 본 개시물의 특정한 양태들이 명료함을 위해 단일 모듈 또는 유닛에 의해 수행되는 것으로 설명되어 있지만, 본 개시물의 기법들은 비디오 코더에 연관된 유닛들

또는 모듈들의 조합에 의해 수행될 수도 있다는 것이 이해되어야 한다.

[0238] 본 개시물의 특정 양태들은 예시의 목적으로 HEVC 표준에 대해 설명되었다. 그러나, 본 개시물에서 설명된 기법들은, 아직 개발되지 않은 다른 표준 또는 사유의 비디오 코딩 프로세스들을 포함하는 다른 비디오 코딩 프로세스들에 대해 유용할 수도 있다.

[0239] 위에서 설명된 기법들은 비디오 인코더 (20) (도 1 및 도 2) 및/또는 비디오 디코더 (30) (도 1 및 도 3)에 의해 수행될 수도 있으며, 그것들의 양쪽 모두는 비디오 코더라고 일반적으로 지칭될 수도 있다. 비슷하게, 비디오 코딩은 해당되는 경우 비디오 인코딩 또는 비디오 디코딩을 지칭할 수도 있다.

[0240] 본 개시물에서 설명된 기법들은 팔레트 코딩의 하나 이상의 상이한 양태들의 다양한 조합들에 대한 기법들을 포함할 수도 있다.

[0241] 하나 이상의 예들에서, 설명된 기능들은 하드웨어, 소프트웨어, 펌웨어, 또는 그것들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현된다면, 그 기능들은 하나 이상의 명령들 또는 코드로서 컴퓨터 관독가능 매체 상에 저장되거나 또는 그것을 통해 송신될 수도 있고 하드웨어 기반 프로세싱 유닛에 의해 실행될 수도 있다.

컴퓨터 관독가능 매체들은, 데이터 저장 매체들과 같은 유형의 (tangible) 매체에 대응하는 컴퓨터 관독가능 저장 매체들, 또는 예컨대 통신 프로토콜에 따라 한 장소에서 다른 장소로 컴퓨터 프로그램의 전달을 용이하게 하는 임의의 매체를 포함하는 통신 매체들을 포함할 수도 있다. 이런 방식으로, 컴퓨터 관독가능 매체들은 일반적으로 (1) 비일시적인 유형의 컴퓨터 관독가능 저장 매체들 또는 (2) 신호 또는 반송파와 같은 통신 매체에 해당할 수도 있다. 데이터 저장 매체들은 본 개시물에서 설명된 기법들의 구현을 위한 명령들, 코드 및/또는 데이터 구조들을 취출하기 위해 하나 이상의 컴퓨터들 또는 하나 이상의 프로세서들에 의해 액세스될 수 있는 임의의 이용가능 매체들일 수도 있다. 컴퓨터 프로그램 제품은 컴퓨터 관독가능 매체를 포함할 수도 있다.

[0242] 비제한적인 예로, 이러한 컴퓨터 관독가능 저장 매체들은 RAM, ROM, EEPROM, CD-ROM 또는 다른 광 디스크 스토리지, 자기 디스크 스토리지, 또는 다른 자기 저장 디바이스들, 플래시 메모리, 또는 소망의 프로그램 코드를 컴퓨터에 의해 액세스될 수 있는 명령들 또는 데이터 구조들의 형태로 저장하는데 사용될 수 있는 임의의 다른 매체를 포함할 수 있다. 또한, 임의의 접속이 컴퓨터 관독가능 매체로 적절히 칭해진다. 예를 들어, 명령들이 웹사이트, 서버, 또는 다른 원격 자원으로부터 동축 케이블, 광섬유 케이블, 연선 (twisted pair), 디지털가입자 회선 (DSL), 또는 무선 기술들 이를테면 적외선, 라디오, 및/또는 마이크로파를 이용하여 송신된다면, 동축 케이블, 광섬유 케이블, 연선, DSL, 또는 적외선, 라디오, 및 마이크로파와 같은 무선 기술은 매체의 정의에 포함된다. 그러나, 컴퓨터 관독가능 저장 매체들 및 데이터 저장 매체들은 커넥션들, 반송파들, 신호들, 또는 다른 일시적인 매체들을 포함하지 않지만, 대신 비일시적, 유형의 저장 매체들을 지향하고 있음이 이해되어야 한다. 디스크 (disk 및 disc) 는 본원에서 사용되는 바와 같이, CD (compact disc), 레이저 디스크 (laser disc), 광 디스크 (optical disc), DVD (digital versatile disc), 플로피 디스크 (floppy disk) 및 블루레이 디스크 (Blu-ray disc) 를 포함하는데, 디스크 (disk) 들은 보통 데이터를 자기적으로 재생하지만, 디스크 (disc) 들은 레이저들로써 광적으로 데이터를 재생한다. 상기한 것들의 조합들은 또한 컴퓨터 관독가능 매체들의 범위 내에 포함되어야 한다.

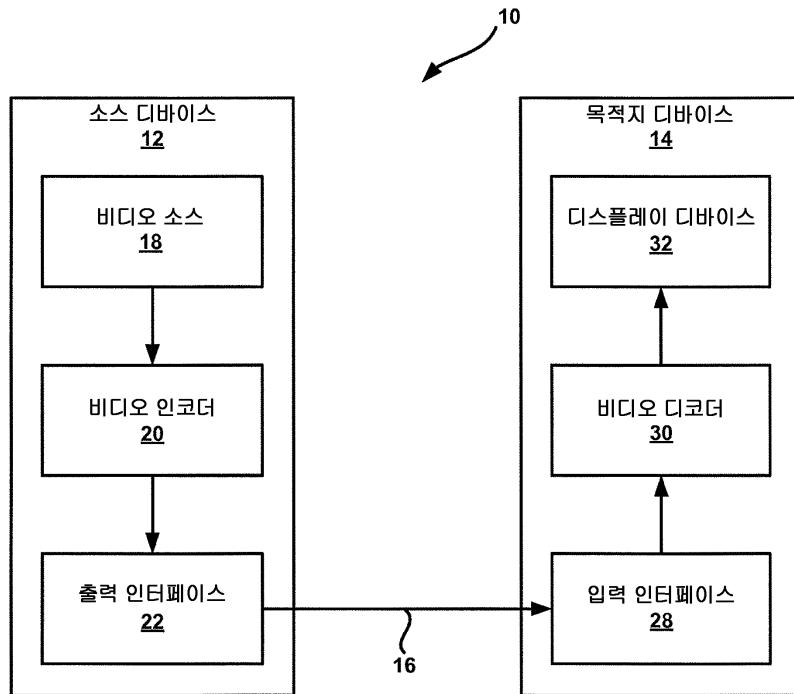
[0243] 명령들은 하나 이상의 프로세서들, 이를테면 하나 이상의 디지털 신호 프로세서들 (DSP들), 범용 마이크로프로세서들, 주문형 집적회로들 (ASIC들), 필드 프로그램가능 게이트 어레이들 (FPGA들), 또는 다른 동등한 집적 또는 개별 로직 회로에 의해 실행될 수도 있다. 따라서, 본원에서 사용되는 바와 같은 "프로세서"라는 용어는 앞서의 구조 또는 본원에서 설명된 기법들의 구현에 적합한 임의의 다른 구조 중 임의의 것을 나타낼 수도 있다. 덧붙여서, 일부 양태들에서, 본원에서 설명된 기능성은 인코딩 및 디코딩을 위해 구성되는, 또는 결합형 코덱 (codec) 으로 통합되는 전용 하드웨어 및/또는 소프트웨어 모듈들 내에 제공될 수도 있다. 또한, 본 기법들은 하나 이상의 회로들 또는 로직 엘리먼트들 내에 완전히 구현될 수 있다.

[0244] 본 개시물의 기법들은 무선 핸드셋, 집적회로 (IC) 또는 IC들의 세트 (예컨대, 칩 셋) 를 포함하는 매우 다양한 디바이스들 또는 장치들로 구현될 수도 있다. 다양한 컴포넌트들, 모듈들, 또는 유닛들이 개시된 기법들을 수행하도록 구성된 디바이스들의 기능적 양태들을 강조하기 위해 본 개시물에서 설명되지만, 상이한 하드웨어 유닛들에 의한 실현을 반드시 요구하지는 않는다. 대신에, 위에서 설명된 바와 같이, 다양한 유닛들은 코덱 하드웨어 유닛에 결합되거나 또는 적합한 소프트웨어 및/또는 펌웨어와 함께, 위에서 설명된 바와 같은 하나 이상의 프로세서들을 포함하는, 상호운용적 하드웨어 유닛들의 컬렉션에 의해 제공될 수도 있다.

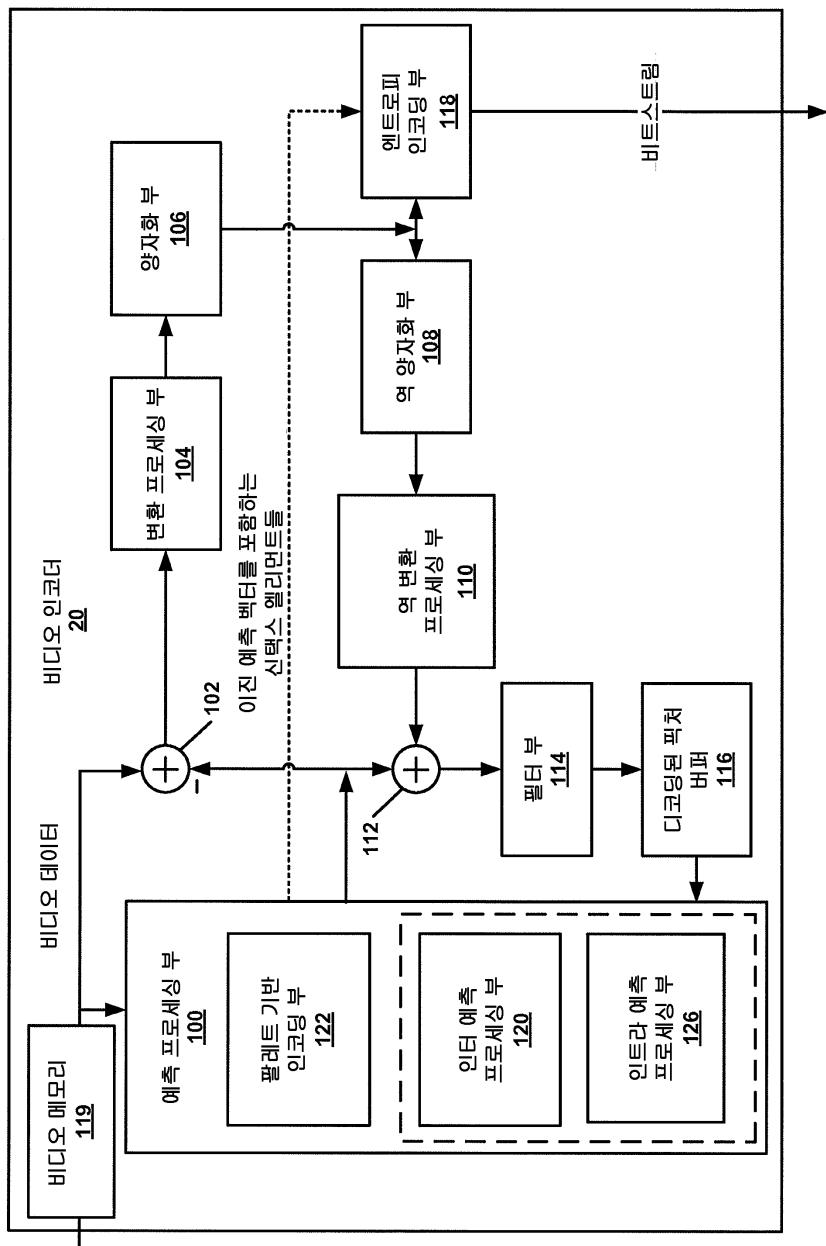
[0245] 다양한 예들이 설명되어 있다. 이를 및 다른 예들은 다음의 청구항들의 범위 내에 있다. 위의 방법들은 개별적으로 또는 조합하여 사용될 수 있다.

도면

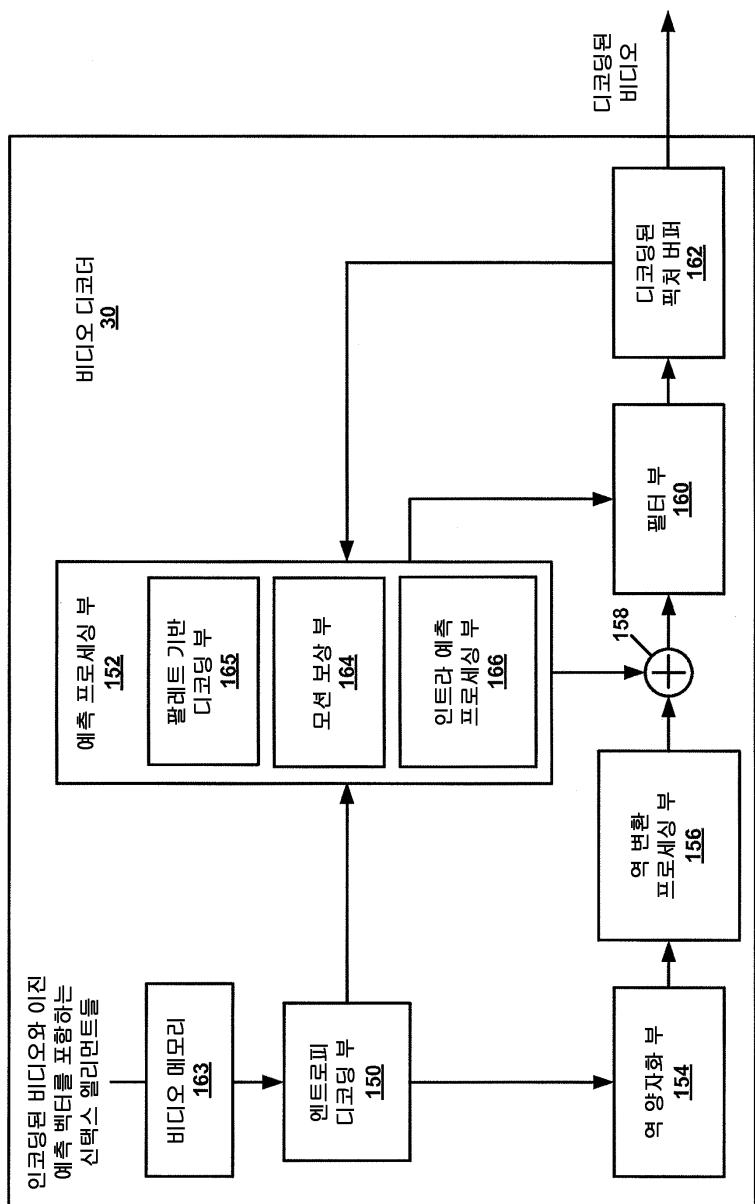
도면1



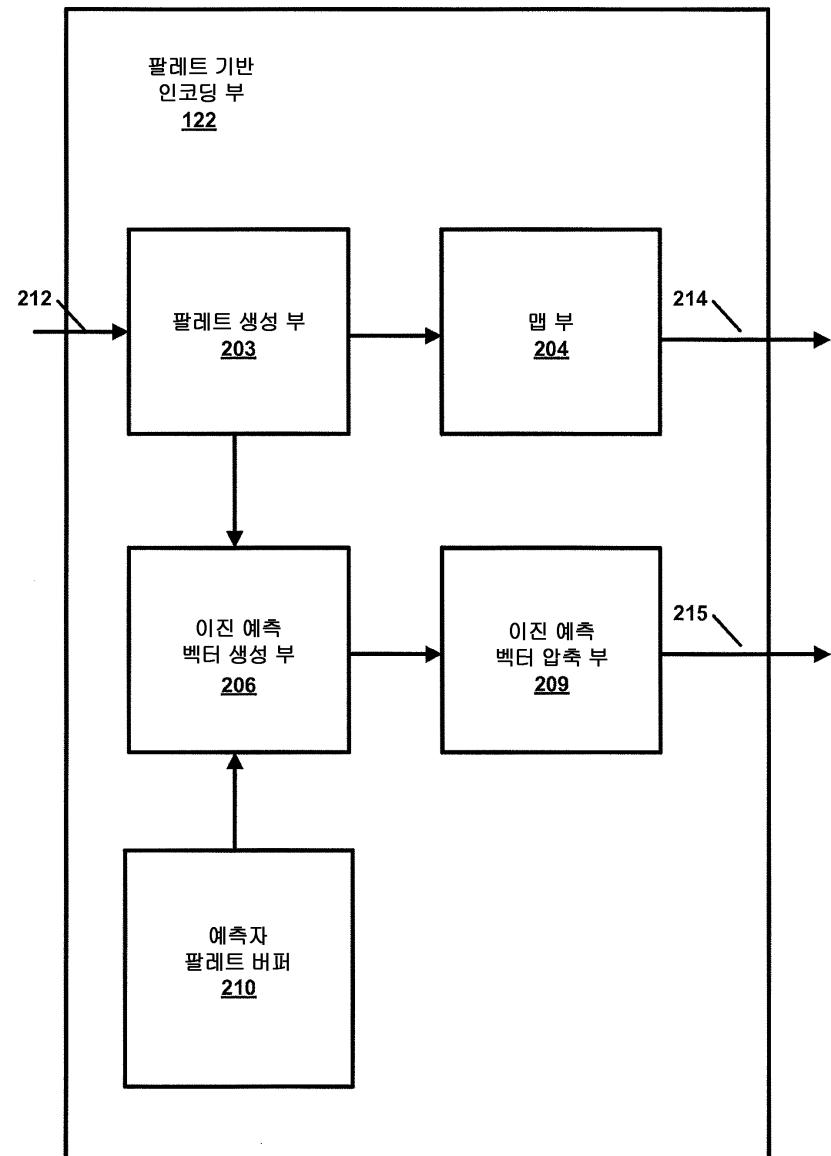
도면2



도면3



도면4



도면5

예측자 플레트 버퍼
210

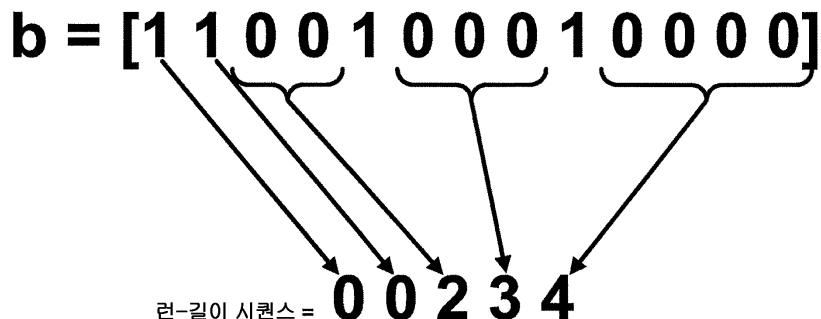
엔트리 인덱스	화소 값
1	값 A
2	값 B
3	값 C
4	값 D
5	값 E
6	값 F
7	값 G
8	값 H
9	값 I
10	값 J
11	값 K
12	값 L
13	값 M
...	...

현재 플레트
220

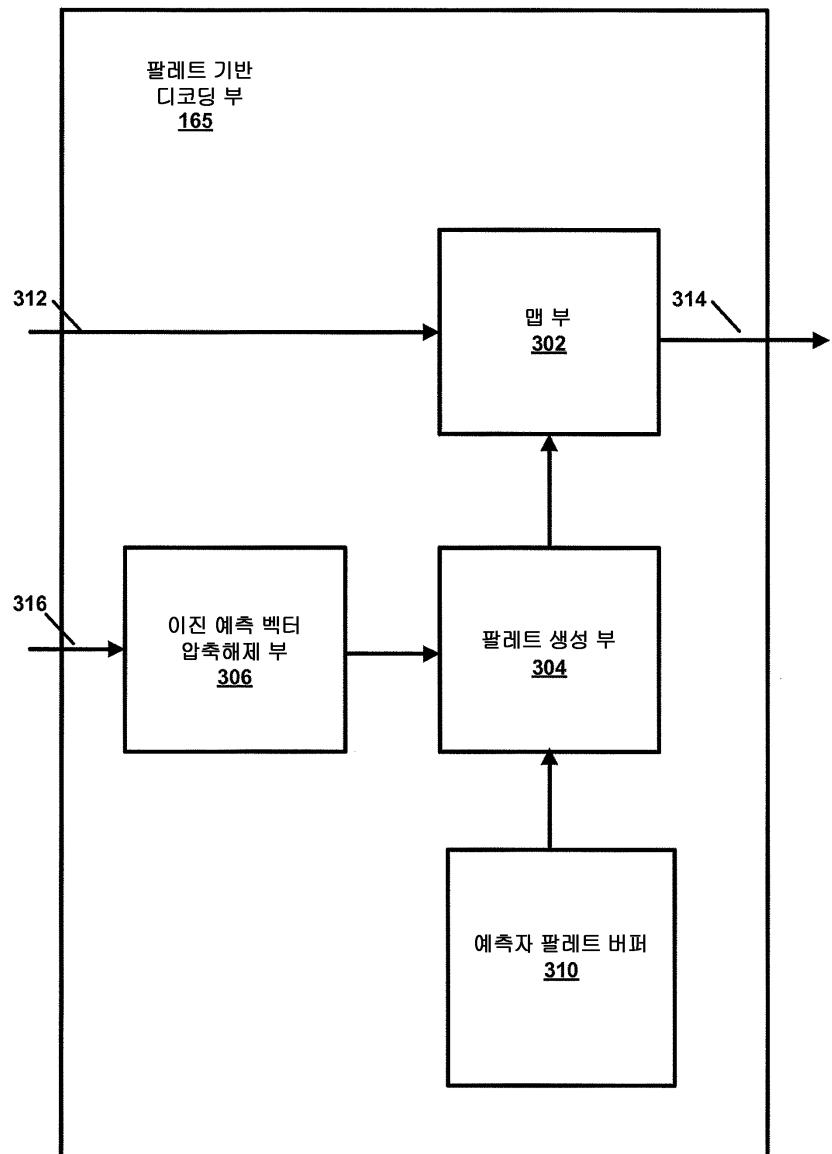
엔트리 인덱스	화소 값
1	값 A
2	값 B
3	값 E
4	값 I
5	새로운 값
6	새로운 값
7	새로운 값
8	새로운 값

$$\mathbf{b} = [1100100010000]$$

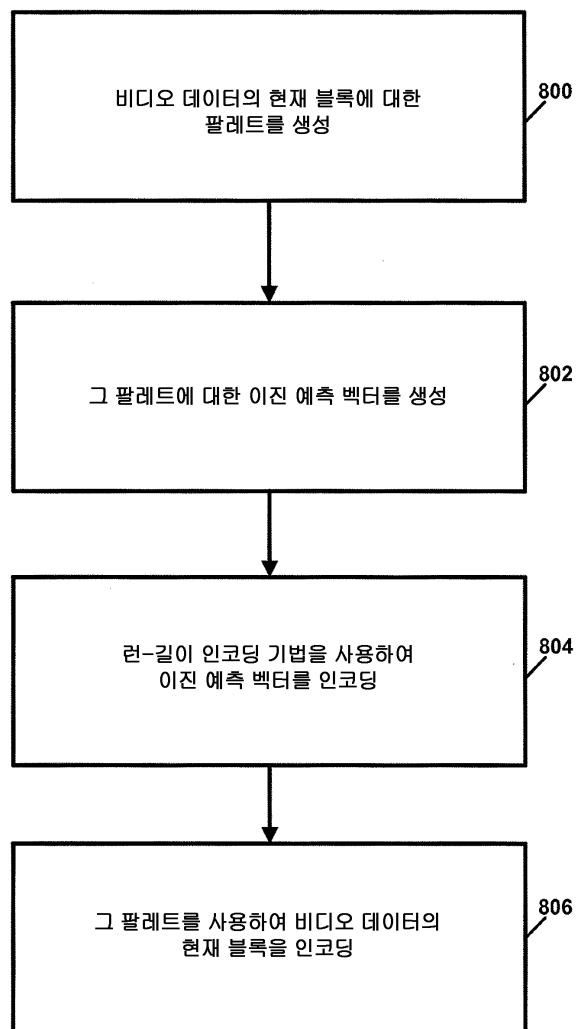
도면6



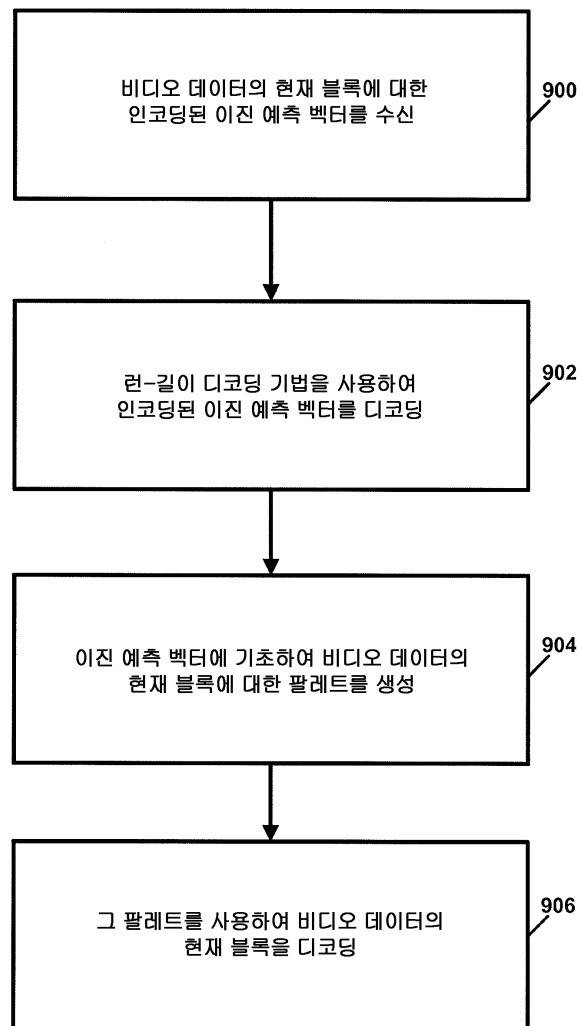
도면7



도면8



도면9



도면10

